CRWR Online Report 98-2

Fugitive Emissions of VOCs from Industrial Sewer Networks: Integration of naUTilus and ArcView

by

Cindy Fee Ha How, M.S.E.

Graduate Research Assistant

and

David R. Maidment, PhD.

Richard L. Corsi, PhD.

Principal Investigators

May 1998

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research • The University of Texas at Austin
J.J. Pickle Research Campus • Austin, TX 78712-4497

This document is available online via World Wide Web at
http://www.ce.utexas.edu/centers/crwr/reports/online.html

Fugitive Emissions of VOCs from Industrial Sewer Networks: Integration of naUTilus and ArcView

by

Cindy Fee Ha How, B.S.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 1998

Fugitive Emissions of VOCs from Industrial Sewer Networks:

Integration of naUTilus and ArcView

Approved by
Supervising Committee:

Richard L. Corsi

David R. Maidment

Howard M. Liljestrand

Acknowledgements

I would like to thank my co-advisors, Dr. Richard Corsi and Dr. David Maidment, for their support and guidance throughout this project. Both have taught me a great deal during the course of my years at The University of Texas at Austin. I would also like to thank Dr. Howard Liljestrand for his guidance in writing this thesis.

I am indebted to David Olson for his patience in instructing me on conceptual and technical aspects of the naUTilus model.

I would like to acknowledge Manuel Cano, of the Shell Development Company, for providing me with data and information used in this research.

This research was made possible through fund by the Environmental Solutions Program at The University of Texas at Austin.

For their support, humor, and understanding, I would like to thank the friends I have made in the EWRE program, with a special mention of two subsets of that group: Dr. Maidment's research group and Team Corsi.

Finally, I would like to extend thanks to my parents, George and Jeannette, and my brother Robin for, simply put, everything. Whether halfway across the world or just across the room, their love and support have been a constant in my life.

May 8, 1998

Abstract

Fugitive Emissions of VOCs from Industrial Sewer Networks: Integration of naUTilus and ArcView

Cindy Fee Ha How, M.S.E.
The University of Texas at Austin, 1998

Co-Supervisors: Dr. Richard L. Corsi and Dr. David R. Maidment

The adoption of the 1990 Clean Air Act amendments brought increasing regulatory attention to on-site industrial sewers as a source of volatile organic compound (VOC) emissions. The goal of this research was to link the FORTRAN model, naUTilus, to GIS technology in order to facilitate prediction of VOC emissions from large industrial sewer networks. The connection of naUTilus with a GIS software package, ArcView®, was achieved through a series of Avenue scripts. The integrated naUTilus/GIS model was used to predict VOC emissions from actual industrial sewer systems under varying environmental, flow, and sewer conditions. Stripping efficiency was predicted to (1) increase with increasing wind speed, (2) increase with increasing temperature (liquid and ambient), (3) decrease with increasing liquid flow rates, and (4) decrease with an increasing number of sealed drains. The integrated model was also used to analyze emissions estimates on a spatial level. Ventilation patterns assumed in the naUTilus model were found to have a significant effect on predicted emissions.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Problem Statement and Motivation

The Clean Air Amendments of 1990 required the U.S. Environmental Protection Agency (USEPA) to establish National Emission Standards for Hazardous Air Pollutants (NESHAPs) for 189 Hazardous Air Pollutants (HAPs) originating from 174 industrial source categories. One source receiving heightened regulatory attention is on-site industrial sewers, particularly those located in petroleum refineries and chemical manufacturing facilities.

In large or mid-sized petroleum refining facilities and the synthetic organic chemical manufacturing industry (SOCMI), emissions of volatile organic compounds (VOCs) can occur from thousands of process drains, vents, and manhole covers. Thus, rigorous emissions monitoring programs are generally infeasible at these facilities and mathematical models are often used to estimate VOC emissions. Present methods for estimating VOC emissions consist of simple emission factors or conservative mathematical models that may overestimate emissions.

The naUTilus model was developed as an alternative method for estimating VOC emissions from industrial sewers. It is based on fundamental mass transfer and fluid mechanics principles with model parameters stemming from several years of field monitoring and pilot experiments. naUTilus predicts emissions from the point of discharge in specific process units, termed "inside the battery limit" (ISBL) and through the main collection system, termed "outside the battery limit" (OSBL). This was done in two separate FORTRAN modules, one corresponding to each type of unit (ISBL and OSBL).

This project seeks to further enhance the naUTilus model through its integration with Geographic Information Systems (GIS) technology. Integrating naUTilus with GIS offers the ability to analyze VOC emissions on a spatial level and provides an easy to use interface for the model. GIS introduces a spatial aspect that allows the connectivity of sewer elements to be established, reducing front-end demands on the user. Both data entry and output display are done in GIS in a visually oriented environment. The analysis of large industrial sewer networks is also more manageable through the use of GIS.

1.2 OBJECTIVES

While naUTilus has been tested on small hypothetical sewer networks, prior to this research it had not yet been applied to large or actual systems. The primary objective of this research is to link naUTilus to GIS technology in order to facilitate prediction of VOC emissions from large industrial sewer networks. As such, the procedure and tools needed to execute naUTilus through the GIS software package ArcView®, distributed by the Environmental Solutions Research Institute (ESRI), were developed. Other objectives include:

- Examining the effect of environmental and flow conditions (such as ambient wind speed, liquid temperature, and liquid flow rate) on VOC emissions from industrial sewer networks.

- Examining the effect of sewer conditions (particularly the number of sealed drains and sealed drain locations) on VOC emissions from industrial sewer networks.
- Evaluating the results of naUTilus on a spatial level. This is more relevant for OSBL units, as the spatial extent tends to be small in ISBL units and large in OSBL units.

1.3 Scope of Research

The procedure developed for integrating naUTilus and ArcView® includes several steps: digitizing a paper schematic, converting digitized files to GIS files, using a specific file storage system, and applying the ArcView® GIS tool. The tool was developed with the goal of creating an easy to use product that does not require detailed knowledge of GIS or naUTilus.

The developed naUTilus/ArcView® application was used to apply naUTilus to a large ISBL and examining various operating parameters. Some of these parameters include liquid temperature, liquid flow rate, ambient wind speed, and sealed drain placement. The ISBL selected was from an actual industrial sewer network. As nearly as possible, accurate dimensions for the ISBL were input to naUTilus.

The naUTilus/ArcView® application was also applied to an actual OSBL unit. As accurate dimensions of the OSBL unit were not available, typical values were used for many sewer characteristics. The OSBL unit was used to demonstrate the ability of the naUTilus/ArcView® tool to connect ISBL and

OSBL units as well as to demonstrate the visual display of results achieved through GIS.

1.4 OVERVIEW

The estimation of VOC emissions from industrial sewer networks by application of the naUTilus model through a GIS interface is discussed in this thesis. Information on (1) industrial sewer networks, (2) various models used to estimate VOC emissions, and (3) previous work where GIS is integrated with models are provided in Chapter 2. The process used to represent and analyze an industrial sewer network in GIS is described in Chapter 3. The results of applying naUTilus to portions of two industrial sewer networks are described in Chapter 4. Conclusions reached through this research, as well as related future work, are discussed in Chapter 5. All scripts, user documentation, and details on hypothetical input created for this research are included in Appendices.

Chapter 2: Background

2.1 INDUSTRIAL SEWER NETWORKS

Industrial sewers exist in every refinery or chemical manufacturing facility as a means of transporting waste streams from process units to an on-site treatment facility or directly to a public sewer. These on-site sewer networks consist of several features: drains, hardpipe connections, cleanout connections, reaches, junctions, drops, and manholes. Waste streams enter the sewer network at drains within a specific process unit or at hardpipe connections. Flow is conveyed toward the treatment facility through a series of pipes (reaches), with a junction where one or more reaches meet. Drops exist at junctions where the out-flowing branch is at a lower elevation than the in-flowing branches. When this configuration occurs, it is often termed a junction box (Howle and Zukor, 1994). Manholes are placed at junctions or along reaches, allowing entry to the sewer system for maintenance and repair purposes. Cleanout connections also exist for maintenance purposes. These are generally closed and have no flow. An illustration of several important sewer elements is shown in Figure 2.1.

Figure 2.2 and Figure 2.3 show the contrast between several types of drains. Figure 2.2 shows an unsealed drain and a sealed drain. Figure 2.3 shows an online versus an elbow drain. At unsealed drains, both air and liquid enter the underlying system of pipes. Sealed drains are sealed with a trap (e.g., P-trap) so air does not enter the system. The difference between online and elbow drains is determined by placement. Elbow drains occur at pipe ends where flow initiates.

Online drains occur at points within the network of pipes, where flow from other parts of the sewer network may join the discharge to the drain. As previously mentioned, flow also enters the system through hardpipe connections. While flow to drains falls from the point of discharge to the underlying sewer system, flow from hardpipe connections enters at the level of existing flow. Liquid does not, however, flow into the system from every drain, as some drains may not be active (i.e. receiving process discharge).

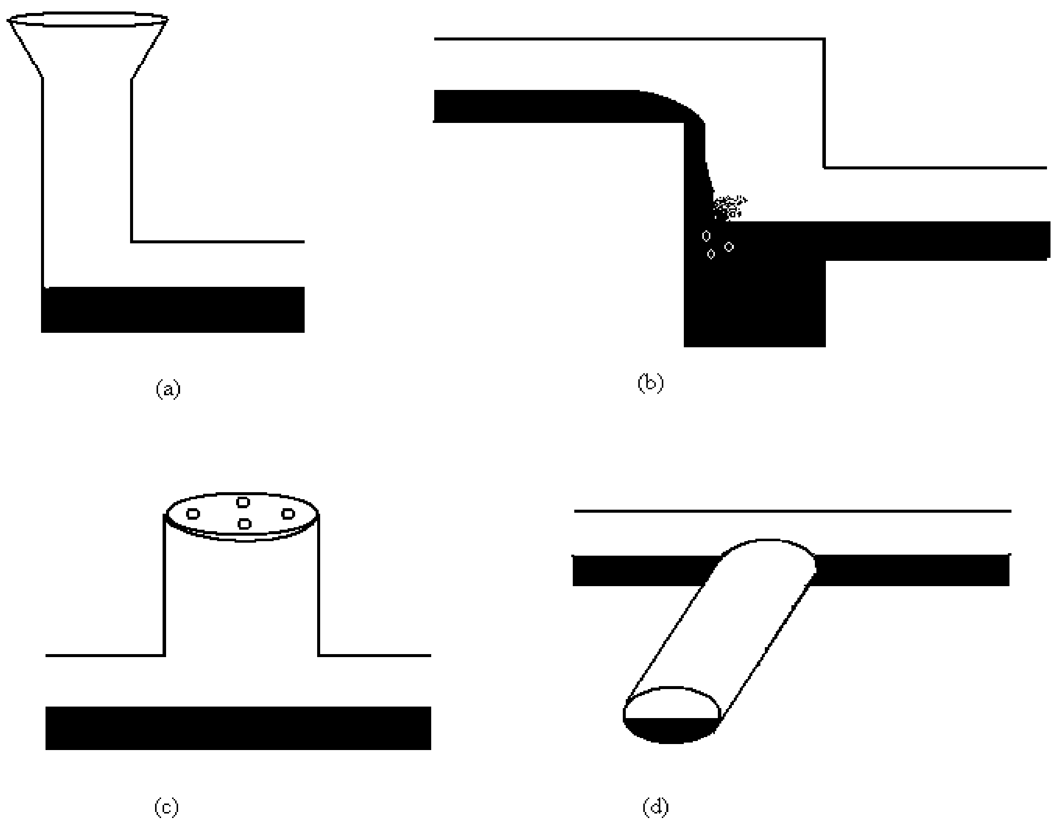


Figure 2.1: Various sewer elements: drain (a), drop (b), manhole (c) and junction (d).

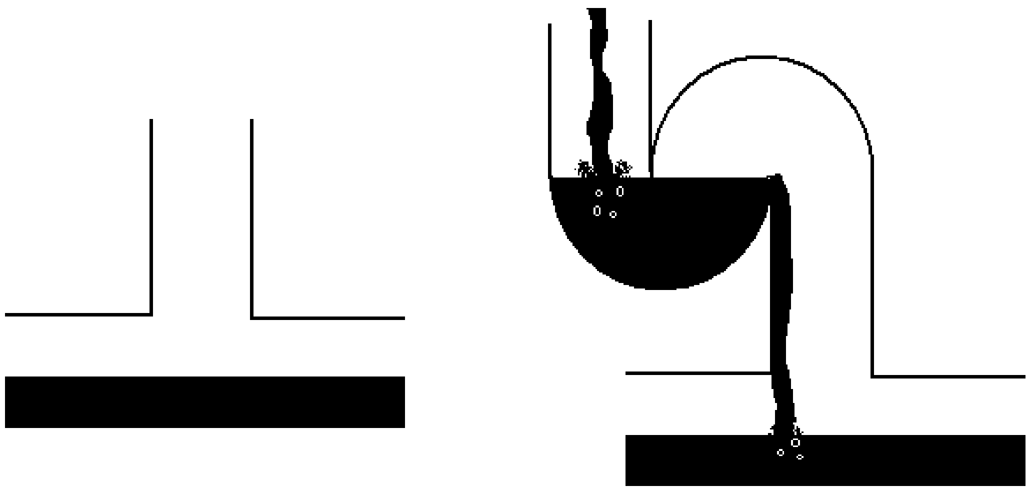


Figure 2.2: Two types of drains: (a) unsealed, (b) sealed (P-trap).

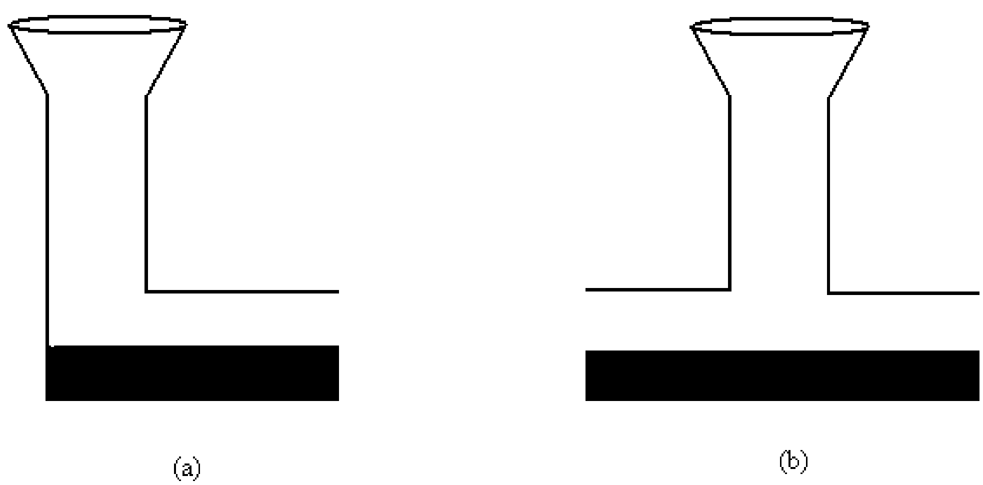


Figure 2.3: Two types of drains: elbow (a) and online (b).

Liquid discharged to sewer lines through drains and hardpipe connections within specific process units, described as "inside the battery limit" (ISBL), is conveyed to larger sewer mains connecting several ISBLs. These main sewer lines comprise the network described as "outside the battery limit" (OSBL). Figure 2.3 illustrates the ISBL/OSBL connection. Several drains feed to a single ISBL unit, with gas-liquid mass transfer occurring at the drains and along the sewer reaches. Air exchange between the sewer and ambient atmosphere occurs at drains, manholes, and vents throughout the ISBL unit. The liquid discharged from the ISBL unit feeds to the OSBL unit. Air exchange between ISBL and OSBL units is inhibited by water seals, placed to prevent the propagation of explosions between process units.

Flow entering the OSBL from ISBL units or other sources contributes to emissions from the OSBL unit. Gas-liquid mass transfer occurs along reaches and at drop structures. Air exchange occurs at openings (manhole covers and elevated vents). A further description of sewer elements is found in EPA (1992) and Howle and Zukor (1994).

2.2 Factors Affecting VOC Emissions from Industrial Sewer Networks

Once discharge to an on-site sewer occurs, two processes contribute to VOC emissions from industrial sewers: mass transfer and air exchange. Several factors affect VOC emissions from industrial sewer networks by changing the amount of mass transfer between the air and liquid phase or changing the air exchange rate in the system. The processes of mass transfer and air exchange, as well as factors that affect those processes, are discussed below.

2.2.1 Mass Transfer

Mass transfer between the liquid and air phases in a sewer system can be estimated using an equilibrium-based system or a kinetics-based approach. A system in equilibrium is a valid assumption for systems with low ventilation or for chemicals of low volatility, where the liquid and gas phase are in contact long enough for equilibrium to occur. In an equilibrium-based system, emissions are calculated as:

$$E = Q_g C_l H_c \tag{2-1}$$

where

E = emission rate (mg/s)

Q_g = air flowrate (m^3/s)

C_l = chemical concentration in the liquid phase (mg/m^3)

H_c = Henry's law constant (m^3_{liq}/m^3_{gas})

In an equilibrium-based system, emissions are directly proportional to air exchange (ventilation rates) and Henry's law constant. Both of these parameters increase as the temperature of wastewater increases.

For chemicals of higher volatility, equilibrium between the gas and liquid phases may not have time to occur and an assumption of equilibrium would cause an overestimation of emissions. These cases are best modeled as kinetics-limited systems. In kinetics-based systems at steady-state, emissions are calculated as:

$$E = K_L A\left(C_l - \frac{C_g}{H_c}\right) \qquad (2-2)$$

$$\frac{1}{K_L} = \frac{1}{k_l} + \frac{1}{k_g H_c} \qquad (2-3)$$

where

K_L = overall mass transfer coefficient (m/s)

A = surface area between liquid and gas (m^2)

C_g = chemical concentration in the gas phase (mg/m^3)

C_l = chemical concentration in the liquid phase (mg/m^3)

k_l = liquid-phase mass transfer coefficient (m/s)

k_g = gas-phase mass transfer coefficient (m/s)

The second term in Equation 2-2 characterizes the concentration driving force, with the term C_g/H_c representing a liquid concentration at which the gas phase would be in equilibrium with the liquid phase.

Factors that affect mass transfer in a kinetics-based system include chemical properties of a compound, temperature, ventilation rate, and hydrodynamics. Some important physical properties are the gas and liquid phase diffusivity (D_g, D_l) and the chemical volatility, which can be described by the Henry's law constant (H_c). Temperature affects mass transfer in two ways; it affects the Henry's law constant, and it affects ventilation in the system. The ventilation rate affects the gas phase concentration in Equation 2-2. For example, an increase in ventilation leads to a decrease in C_g and, hence, an increase in emissions. Hydrodynamic conditions (dictated by liquid flow rate and reach diameters) affect the hydraulic residence time in the system and therefore the time during which mass transfer can occur. Hydrodynamic conditions also have a major effect on liquid-phase mass transfer coefficients, increasing with increases in the magnitude of total kinetic energy in water. Aerodynamic conditions have similar effects on the gas-phase mass transfer coefficients.

2.2.2 Air Exchange

The magnitude of air exchange plays a large role in the extent of mass transfer. Air exchange, as described by a ventilation rate, is driven by several processes: liquid drag, buoyancy, wind eduction, rise and fall of wastewater level, and barometric pressure changes (Olson *et al.*, 1997a).

Olson *et al.* (1997a) discuss the various processes that influence ventilation. Liquid drag induces ventilation by drawing air into a sewer system when liquid enters the system at an unsealed drain. It is caused by shear forces at the air-water interface. A buoyant effect induces ventilation through temperature differences between ambient air and air in the sewer headspace. This temperature difference creates density differences, causing the hotter, buoyant air in the sewer to rise. Wind eduction is caused by differential wind speeds across sewer openings. The wind causes dynamic pressure differences leading to air inflow and outflow. Rising and falling wastewater levels can force air out of the sewer system. Variations in barometric pressure lead to expansion and compression of gases, also forcing air in and out of the system.

2.3 METHODS FOR ESTIMATING VOC EMISSIONS

Some methods for estimating fugitive emissions from industrial sewer networks are discussed in a report to the American Petroleum Institute (API, 1996). Most methods for estimating VOC emissions from industrial sewer networks are based on emission factors and mathematical models. The simplest method for estimating VOC emissions from industrial sewer networks involves the use of an AP-42 emission factor. This method assumes that all drains emit at the same rate, with a given quantity (0.07 lb/hr) released to the atmosphere per drain. The influence of process parameters such as temperature and reactant concentration are not accounted for in the AP-42 emission factor (USEPA, 1995).

The best available control technology/lowest achievable emission rate (BACT/LAER) approach is based on emission factors, taking effects of air flow rate, water flow rate, and the Henry's law constant into consideration (USEPA, 1990). Some assumptions made for BACT/LAER calculations for wastewater collection systems are:

- Design depth in the channel is half full.
- Flow in the channel for estimating fractional emissions is 80% of the design depth.
- Air exiting the system is in equilibrium with the wastewater concentration.
- Wind speed is 3.5 mph.

Emissions estimates can be done with BACT/LAER through spreadsheets, with emissions factors for various sewer elements. An example calculation of emissions using BACT/LAER emission factors is shown in Table 2.1. AP-42 and BACT/LAER calculations are the simplest methods for estimating VOC emissions from industrial sewers. They require the least input and do not require a description of the specific sewer configuration.

Table 2.1: Example of BACT/LAER emissions calculations (taken from USEPA, 1990)

Unit	**Emission Factor**	**Amount Present in liquid (g)**	**Emissions (g)**
Open trench drain (40 ft)	0.045	100	4.5
Open trench drain (20 ft)	0.022	95.5	2.1
Drain	0.08	93.4	7.5
Drain connection	0.08	85.9	6.7
Manhole at junction	0.0083	79.2	0.66
Covered sump with vent	0.11	78.5	8.6
Overall collection units	0.30	70	30

The Wastewater Treatment Compound Property Processor and Air Emissions Estimator program (WATER8) is a model based on BACT/LAER calculations. It uses the same assumptions listed in the description of the BACT/LAER calculations, and it consists of analytical expressions for emissions estimates that are based on chemical properties and concentrations in the waste streams (USEPA, 1994b). Emission factors developed in WATER8 are reported in terms of the fraction of chemical mass emitted per wastewater collection or treatment unit (USEPA, 1994a). These emission factors are based on various cases representing sewer elements under different conditions. For example, Table 2.2 shows an example of emission factors for 1,3-butadiene. The typical value used for fractional emissions is the average of the three cases. WATER8 is a menu-driven program executed from a DOS platform. It is distributed by the USEPA through the Office of Air Quality Planning & Standards and can be readily downloaded on the internet (http://www.epa.gov/ttnchie1/software.html).

Table 2.2: Emissions factors for 1,3-butadiene listed as fraction emitted. Each of the three cases represents a different airflow condition. From USEPA(1994a).

Case	Drains (A)	Manholes (B)	Collection Conduits (C)
Case 1	0.63	0.087	0.95
Case 2	0.73	0.21	0.79
Case 3	0.54	0.147	0.56
Typical value	0.63	0.15	0.77

The Collection system Organic Release ALgorithm (CORAL), developed by Corsi *et al.* (1992) and later revised to CORAL+, models VOC emissions from sewer reaches. The model divides each reach into a series of continuous-flow stirred-tank reactors (CFSTRs), and performs a mass balance on the liquid and gas phases in the system. Data required by the algorithm include physical and flow characteristics of the sewer, ventilation rate, and compound properties. Results from the model compared favorably with data collected from field experiments in municipal sewers (Corsi, *et al*., 1992; Whitmore and Corsi, 1994). A limitation to CORAL+ is the model's lack of an algorithm for modeling emissions from drains or drop structures.

The Secondary Emission Assessment Model (SEAM) program represents reaches as a series of CFSTRs. This model, developed and distributed by ENVIROMEGA, Ltd., is a Windows®-based program in which the user inputs the sewer network by defining sewer elements and sets various parameters used in calculations. SEAM uses mass transfer principles and calculates a mass balance throughout the system. An amount of volatilization is calculated for drains, drop structures, surfaces (in reservoirs), and sewer reaches. Volatilization calculations are based on various previously published correlations (Nakasone, 1986; Parkhurst and Pomeroy, 1972; Mackay and Yeun, 1983). SEAM also allows for sorption of chemicals onto a solid phase. Once SEAM has been executed, emission "hot spots" can be displayed.

The SEAM model also has limitations, such as the lack of a ventilation calculation. The user must input a ratio, Q_g/Q_l, to specify the ventilation rate. The user must also specify a number of CFSTRs to be used in modeling concentrations in reaches. The model also requires each drain to have non-zero inflow.

2.3 The naUTilus model

The naUTilus model was developed at the University of Texas at Austin under the supervision of Dr. Richard Corsi. The model predicts the extent of gas-liquid mass transfer and the amount of air exchange, giving an estimation of both the level of emissions from ISBL units, as well as the concentration and flow rate of discharge to a corresponding OSBL unit. Further, naUTilus estimates the amount of mass transfer and air exchange in the OSBL unit.

The naUTilus model is based on fundamental mass transfer principles for estimating gas-liquid mass transfer. Fluid mechanic and heat transfer principles are used for estimating air exchange rates between the sewer and the ambient atmosphere. The specific mechanisms of mass transfer and air exchange used as a basis for the algorithms in naUTilus are both theoretically and experimentally based. They describe the physical situations occurring at or along each of the sewer features as discussed above. A comprehensive description of the naUTilus model is provided in naUTilus documentation (Olson *et al.*, 1997b). The following sections summarize some important aspects of the naUTilus model

2.3.1 naUTilus File Structures

Due to differences between ISBL and OSBL units, naUTilus uses two separate FORTRAN modules, each run independently. The ISBL module of naUTilus reads data from an input file (ISBL.IN) that describes the inflows to an ISBL unit, as well as drain characteristics, branch characteristics, manhole and junction locations, and branch/node connectivity of the unit. Chemical property data must also be included in the input file. The ISBL naUTilus module outputs data into two files: ISBL.OUT and ISBLOUT.TXT. The ISBL.OUT file is a text file that lists liquid flow rate, water temperature, air flow rate, liquid concentration, and gas concentration for each branch in the ISBL unit. It also summarizes the total emissions from the ISBL unit. The ISBLOUT.TXT file lists the aforementioned emission rate, total mass of chemical entering the ISBL unit, emission rate from the ISBL unit, liquid flow rate, liquid concentration, liquid temperature, and oil fraction of the flow entering the OSBL from the modeled ISBL unit. Example ISBL input and output files are included in Appendix D.

The OSBL module of naUTilus reads data from an input file (OSBL.IN) that OSBL.IN is very similar, but not identical, to the ISBL input file. The OSBL input file describes the flows entering the OSBL unit from ISBL units or other sources, in contrast to the drains described in the ISBL input file. Like the ISBL input file, the OSBL input file also describes manhole and junction locations, branch diameters and slopes, and branch/node connectivity. The OSBL naUTilus module outputs data into three files: OSBL.OUT, ONDOUT.TXT, and OBROUT.TXT. The OSBL.OUT is a text file listing liquid flow rates, liquid temperatures, and chemical concentrations for each branch in the OSBL unit. It also lists gas flow rates and mass emissions from each node in the unit. OSBL.OUT includes summary information on the total mass input to the system, total emissions, and overall stripping efficiency. ONDOUT.TXT also lists the nodal emissions and gas flow rates. ONDOUT.TXT is formatted with one line per node, with data separated by columns, to allow the data to be viewed in tabular format when imported to ArcView® or Excel. OBROUT.TXT is formatted in a similar way, with one line for each branch in the system. It holds data on liquid flow rate, liquid temperature, gas flow rate, liquid concentration, and the mass transfer coefficient calculated for each branch. Examples OSBL input and output files are included in Appendix D.

In addition to the differences in the input and output file names and formats, other differences between OSBL and ISBL units motivated the development of separate ISBL and OSBL modules. The differences in mass transfer and air exchange are discussed in the following two sections.

2.3.2 Mass Transfer in naUTilus

As mentioned in the previous section, the naUTilus model considers differences between mass transfer in ISBL and OSBL units. The ISBL module of naUTilus gives the user the options of using a kinetics-limited system or an equilibrium-limited system for mass transfer calculations. The ISBL module must also calculate values for above sewer emissions and mass transfer at drains. Above sewer emissions occur at drains with a water seal where mass transfer occurs between the liquid stream and the ambient air, with the air prevented from entering the sewer system by the water seal.

For OSBL units, which tend to be larger and less ventilated than ISBL units, calculations are done for two scenarios. One scenario assumes that reaches come to gas-liquid phase equilibrium. The second scenario, called the open trench model, assumes that the system is open to the atmosphere and thus has infinite ventilation. For infinite ventilation, mass transfer is kinetics limited. The calculations for both scenarios are compared and the lower of the stripping efficiencies reported for the unit. The lower value is assumed to be valid, as both scenarios are conservative and the combination of the methods is believed to yield less error than either model would predict if used exclusively (Olson *et al.*, 1997b).

2.3.3 Air Exchange in naUTilus

Air exchange in naUTilus is based on the mechanisms described in Section 2.2.2. In addition to several mechanisms contributing to ventilation, ventilation patterns can be quite complex. For both ISBL units and the equilibrium-limited scenario for the OSBL unit, air exchange is an important factor in estimating emissions. Again, methods for determining ventilation in ISBL units and OSBL units vary in the naUTilus modules.

In ISBL units, the amount of ventilation depends greatly on the open/closed status of the system. In open systems (no sealed drains), relatively large air flow rates can exist. The air exchange due to wind eduction is assumed to be at its maximum value when 45% of the openings ingas and the remaining 55% outgas (Varma, 1995). Varma determined the maximum air exchange rate for this condition as:

$$Q_{exit}^{max} = 0.22XAV_w \qquad (2\text{-}5)$$

where

Q_{exit}^{max} = total air flowrate exiting the system due to wind eduction (m^3/s)
X = total number of openings in the system
A = area of each opening (m^2)
V_w = ambient wind velocity (m/s)

This total air flow rate is then evenly distributed among all openings in the ISBL system.

Two forms of buoyancy contribute to ventilation in open ISBL systems: channel buoyancy and process buoyancy. Channel buoyancy occurs when heat transfer between the liquid and air along sewer reaches induces air flow. In naUTilus, air flow caused by channel buoyancy is equally divided among all drains in the system. For process buoyancy, air flows are caused by heat transfer as the liquid falls from the drain to the underlying sewer, with subsequent heat transfer during splashing as the liquid enters the underlying flow. In naUTilus, process buoyancy is assumed to induce an air flow circuit between the drain where flow occurs and the nearest inactive drain (i.e. no flow) or pick hole (Olson *et al.*, 1996). In the naUTilus model, ventilation due to water drag is assumed to be negligible compared to ventilation induced by buoyancy and wind eduction, thus allowing the effects of water drag to be ignored for ISBL calculations on open systems.

In contrast, air exchange in closed systems is relatively low. Air exchange can only occur through pick-holes on manhole covers. The lowered contribution of buoyancy and wind eduction to ventilation makes the contribution of water drag to the total ventilation significant. Ventilation due to water drag is calculated with 45% of the nodes ingassing and the remaining 55% of the nodes outgassing. The ingassing nodes are assumed to be those furthest from the sewer network outlet, consistent with the findings of Quigley and Corsi (1995). Also, closed systems do not exhibit process buoyancy due to the presence of water seals at drains. Channel buoyancy and wind eduction are calculated and distributed in the same manner for a closed ISBL system as in an open ISBL system.

The relevant mechanisms for ventilation in equilibrium-limited OSBL units are wind eduction, water drag, and buoyancy. Ventilation pattern and calculations for OSBL are identical to those used for open ISBL systems for wind eduction. Ventilation in OSBL units due to water drag is calculated using the same method as for closed ISBL systems. Air exchange due to buoyancy in OSBL systems is equally distributed among all OSBL openings. As previously mentioned, the open trench model assumes infinite ventilation, so a ventilation calculation is not necessary.

2.3.4 Other Relationships used in naUTilus

When calculating mass transfer in an ISBL system, naUTilus offers the options of an equilibrium-limited system and a kinetics-limited system. The naUTilus model also offers options for two other calculations: mass transfer at drop structures and mass transfer along reaches. For mass transfer at drop structures, naUTilus does calculations based on an oxygen deficit ratio, employing a relationship developed by Nakasone (1986), or using a general weir model, employing equations used in WATER8.

Calculations for mass transfer along reaches presently employ relationships for mass transfer coefficients developed by Parkhurst and Pomeroy (1972) or Owens *et al.* (1964). Relationships for mass transfer along reaches have recently been developed by Jacek Koziel at the University of Texas at Austin and will shortly be implemented in the naUTilus model (Koziel, 1998). Further descriptions of the relationships used for reaches and drop structures are included in Olson *et al.* (1997b).

The naUTilus model also includes four relationships to determine the Henry's law constant. In method one, the Henry's law constant is a function of temperature calculated using the van't Hoff equation presented by Ashworth *et al.* (1988). The van't Hoff equation is written as:

$$H_c = \exp\left(A - \frac{B}{T}\right) \qquad (2\text{-}6)$$

where

H_c = Henry's law constant (m^3_{liq}/m^3_{gas})

A, B = Empirical constants

T = Temperature (K)

In method two, the Henry's law constant is a function of temperature, adjusted using the ratio of vapor pressures and a known Henry's law constant at a given temperature. Vapor pressure is calculated using Antoine's constants and Equation 2-7. The Henry's law constant is then calculated using equation 2-8.

$$\ln(P) = A - \frac{B}{(T + C)} \qquad (2\text{-}7)$$

where

P = Vapor pressure (mm Hg)

A, B, C = Antoine's constants

T = Temperature (K)

$$H_c = H_{c,T1} \times \frac{P_{T2}}{P_{T1}} \qquad (2\text{-}8)$$

where

$H_{c,T1}$ = Known Henry's law constant at temperature, T_1

P_{T1} = Vapor pressure (mm Hg) at temperature T_1

P_{T2} = Vapor pressure (mm Hg) at temperature T_2

The third method used also uses the Antoine's constants. The Henry's law constant is calculated as the ratio of the vapor pressure to the solubility, as shown in Equation 2-9.

$$H_c = \frac{P}{\text{Sol.}} \qquad (2\text{-}9)$$

where

P = Vapor pressure as calculated using Antoine's constants (converted from mm Hg to g/L)

Sol. = Solubility (g/L)

The final method assumes a single value of Henry's law constant throughout the system. This single value is specified by the user.

2.4 GEOGRAPHIC INFORMATION SYSTEMS

Geographic information systems (GIS) software is a category of software that allows tabular data to be directly linked to a map. As described by ESRI (1998):

> "A geographic information system (GIS) is a computer-based tool for mapping and analyzing things that exist and events that occur on the earth. GIS technology integrates common database operations such as query and

statistical analysis with unique visualization and geographic analysis benefits offered by maps."

GIS is a useful tool that is gaining widespread use for managing and modeling spatial data. It has numerous potential applications and is already applied in several fields. Douglas (1995) lists GIS applications in four categories: (1) government management and planning, (2) environmental safety and health applications, (3) resource planning, and (4) commercial planning. Tasks included under environmental safety and health applications are wastewater management and air emissions.

The applications for industrial facilities are identified as automated mapping and facilities management (AM/FM). Berry (1993) describes AM/FM applications as "descriptive," in contrast with "prescriptive" applications that involve spatial statistics and modeling. Prescriptive applications are identified by the term decision support systems (DSS). Additionally, Nobel (1998) is currently conducting research regarding the application of GIS towards facilitating industrial ecology at Eco-Industrial Parks (EIP) (Nobel, 1998). Nobel's work incorporates optimization models and the ArcView® GIS.

GIS software has three features of particular interest for this research. First, GIS allows a user to view maps and assign data to map features. Each map layer has a feature attribute table, with each feature of a map having a unique entry and identification number (see Figure 2.4). This link between maps and attribute tables allows for display of attributes on the map. In ArcView®, the primary GIS software used in this research, maps are displayed in the View window.

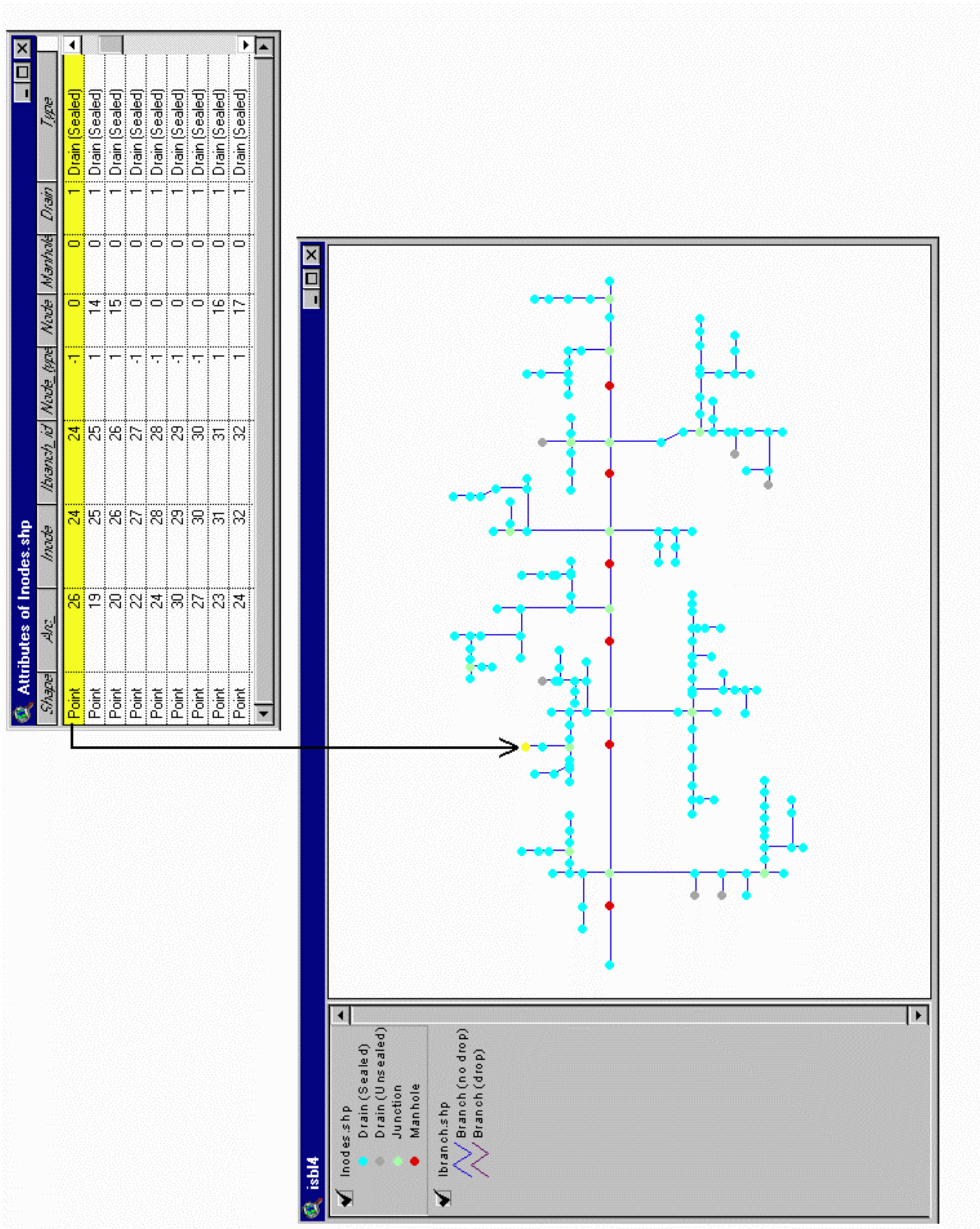


Figure 2.4: Feature attribute table shown with the corresponding map in an ArcView® View window. The selected item on the map corresponds to the selected item in the table.

Second, GIS can also establish connectivity of polygon or line map features. Sewer networks are best represented as a series of lines and points. Reaches are described as lines. Drains, manholes, and junctions are represented by points. Each point is numbered and each line is numbered. Lines are also assigned a "from node" and a "to node," as seen in Figure 2.5. These nodes depict location as well as direction. This feature allows the connectivity of the sewer network to be established without manual effort.

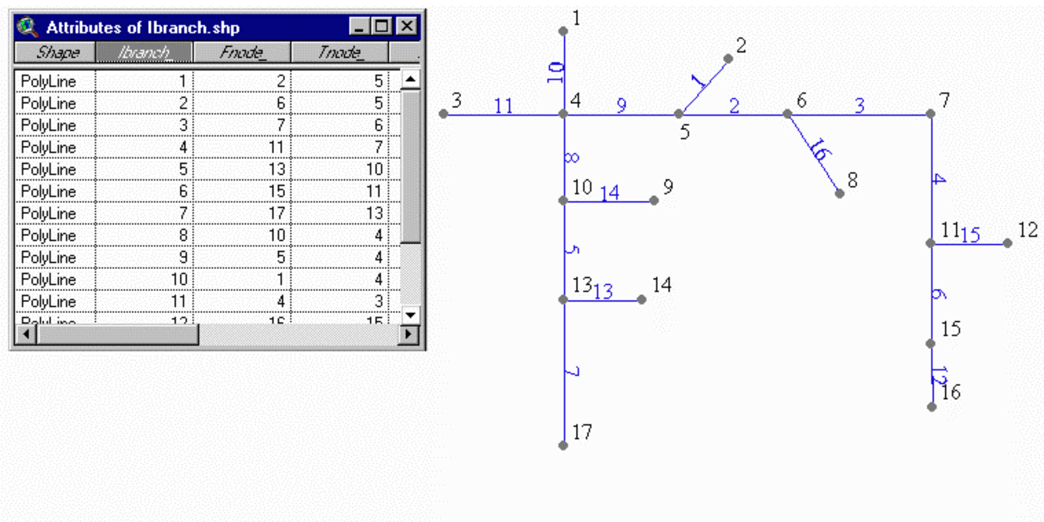


Figure 2.5: Example schematic of an industrial sewer, with a table of attributes for the branches. Nodes are numbered in black and branches are numbered in blue. The "from node" and "to node" for each branch are indicated in the attribute table.

Third, user written programs can customize GIS software to specific applications. These programs eliminate requirements of user expertise in the GIS application. In ArcView®, the user can be prompted for input, input can be processed, and files can be manipulated to read and write necessary information without high demands on the user. ArcView®, distributed by ESRI, is customized through scripts. Scripts are programs written in the object-oriented language Avenue, which operates within ArcView®.

2.5 GIS/MODEL INTEGRATION

Several views and examples exist of the integration of GIS with various models. Tim and Jolly (1994) described three levels of integration with respect to their integration of GIS and the hydraulic/water quality model AGNPS. Figure 2.6 shows the levels of integration. The simplest type of integration is ad hoc integration. In this case, GIS generates data for the model. Model execution and output are independent of the GIS. Partial integration involves input generation in the GIS and model output analysis through the GIS. Complete integration consists of programming the model within the GIS. These levels of integration are also discussed in terms of loose coupling, i.e. ad hoc integration, and tight coupling, i.e., complete integration, by Nyerges (1993).

Benaman (1996) lists many examples of GIS/model integration and notes that most use the GIS package Arc/Info. Noting the current trend towards ArcView® and Avenue, Benaman developed a link between ArcView® and the WASP5 water quality model using partial integration. Models within ArcView® (complete integration) include a surface water quality model that computes a constituent mass balance for surface water segments (Hellweger, 1997), and a surface water flow simulation incorporating both surface water and ground water modeling (Ye, 1996).

In the area of air pollution, a pilot study was conducted in which GIS and the regional oxidant model (ROM) were integrated (Novak and Dennis, 1993). In the pilot study, the Interactive Display for Environmental Analysis System (IDEAS) was developed utilizing pull down menus in Arc/Info. It allowed comparisons between measured and predicted air pollution concentrations, point and area source emissions, land use, and health effects. GIS was found to be a useful means to design emissions control strategies and to estimate input to air quality models.

Maitin and Klaber (1993) integrated GIS and air dispersion modeling using the Industrial Source Complex (ISC2) model. They discuss the value of GIS for improving visual display of input and results, reducing time requirements for data input, querying to determine impacts of altering input parameters, and redefining receptor grids with minimal effort. Maitin and Klaber also include the risk assessment after modeling as a potential use of GIS.

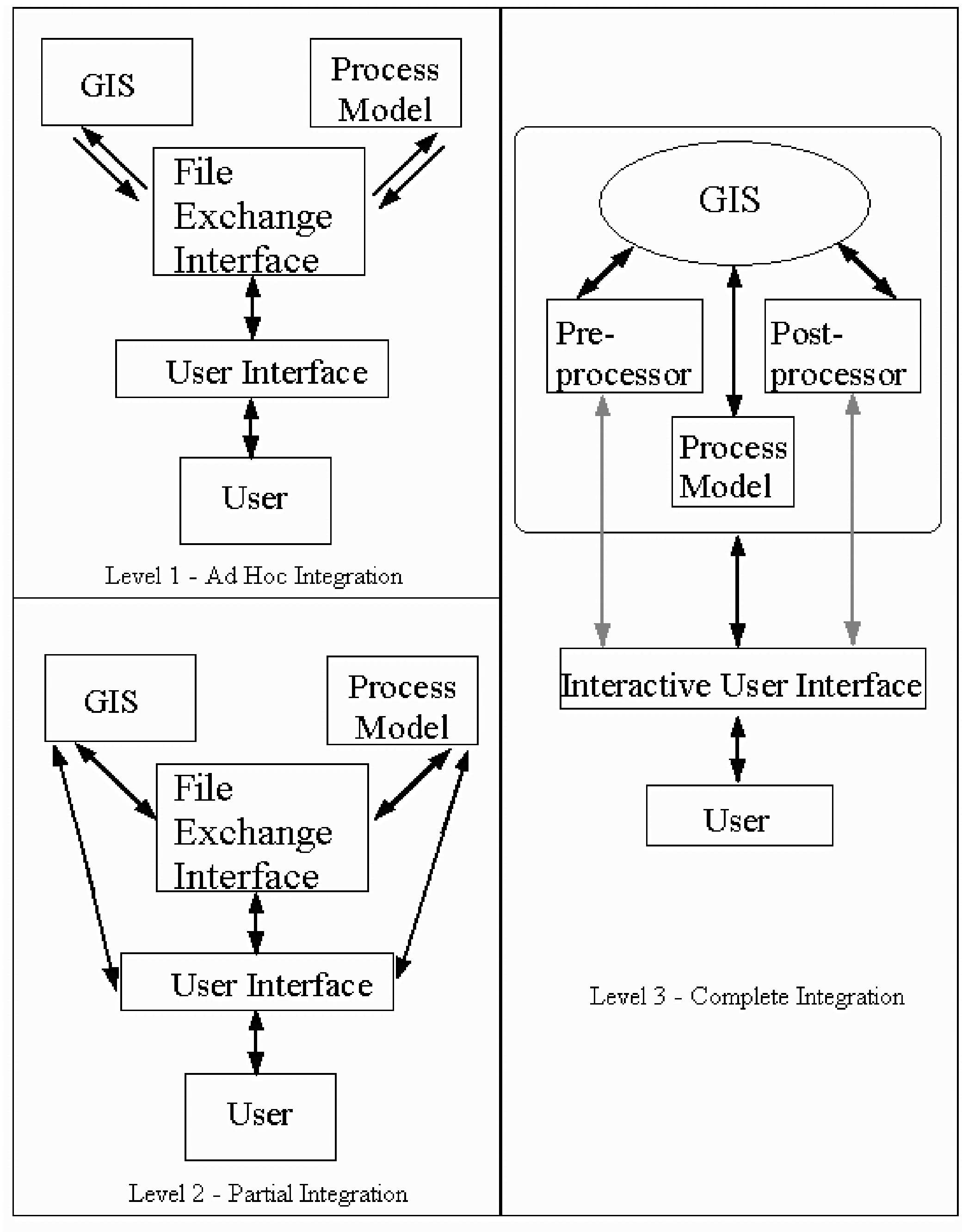


Figure 2.6: Three levels of integration, as described by Tim and Jolly (1994).

Chapter 3: Process Development

The following sections describe the process developed for entering industrial sewer networks in ArcView® GIS, for running the naUTilus model from within ArcView®, and for viewing naUTilus results in ArcView®. Section 3.1 describes typical data sets as well as the specific data sets used to complete the research described herein. Steps designated as "preprocessing" steps, needed to reduce data to a format appropriate for ArcView® are described in Section 3.2. The programs written to connect ArcView® and the naUTilus model are described in Section 3.3.

3.1 DATA SETS

Data generally provided on industrial sewer networks are in the form of schematics or diagrams. The schematics tend to be available only in hard copy and are numbered in several different ways. This discrepancy in numbering systems makes comparison ofdata sets cumbersome. For a discussion of various methods of numbering industrial sewer elements, see Appendix A.

The data sets used in this research were based on parts of actual industrial sewer networks. The schematic and sewer characteristics for an ISBL unit were acquired from the Shell Development Company. The schematic provided (see Figure 3.1) is not to scale, but was used as an accurate representation of the sewer network. It represents an ISBL unit with 97 drains (unsealed), 226 branches, 9 manholes, and 36 cleanout connections. A more accurate plant drawing was later provided, along with tables containing sewer dimensions. The data sets generated from the original schematic were adjusted to accurately represent the dimensions of the sewer network. The ISBL unit is approximately 60 meters by 120 meters (200 ft by 400 ft). Branch lengths originally read from the digitized schematic were not accurate and were later corrected by attributing the proper length to each branch. This was done to ensure consistency with the provided data sets describing the ISBL unit.

The Shell Development Company also provided the schematic for an OSBL unit (see Figure 3.2). The unit is part of a chemical manufacturing facility. It extends approximately 1500 meters by 500 meters. The OSBL has 68 manholes and 15 junctions (no manholes present). Detailed sewer dimensions were not provided, nor were data provided on flow to the system. Values used for the OSBL unit dimensions are considered typical values. Branch diameter values of 0.5 meter and 1 meter were used, with a branch slope of 0.01%.

While schematics were provided, limitations to the data existed. First, although the more accurate version of the ISBL schematic would have been the optimal choice for use throughout the project, it was not made available until most of the work had been completed on the approximate ISBL schematic. This necessitated the adjustments made so that actual dimensions were represented as accurately as possible. When making these adjustments, further errors were observed in the approximate schematic and the files representing the drawings were altered accordingly. The plant drawing, plant locations, and specific dimensions are not included in this report due to their proprietary nature. Second, actual flow data for the ISBL and OSBL units were not available, nor were the

ISBL unit and OSBL unit parts of the same facility. The latter of the listed limitations necessitated the use of hypothetical flow values for both ISBL inflow and OSBL inflow. The hypothetical flow values are further in Chapter 4. Despite these limitations, the ISBL and OSBL units that were used for this study are representative of the complexity of on-site industrial sewers and served to illustrate the application of GIS to such systems.

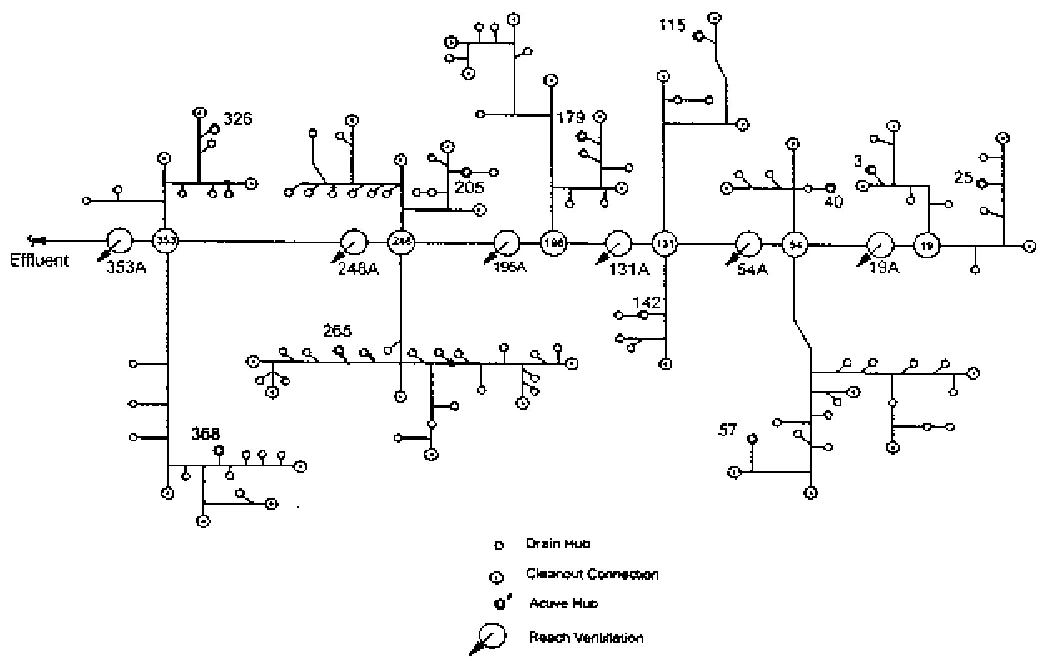


Figure 3.1: ISBL schematic used to create GIS files. The numbers shown are consistent with the numbering scheme used by MW Kellogg (see Appendix A). The numbers along the main sewer line correspond to manholes and reach ventilation. Other numbers correspond to active drains (drains with inflow). Clean out connections were not digitized or otherwise used in this research. (Figure not to scale).

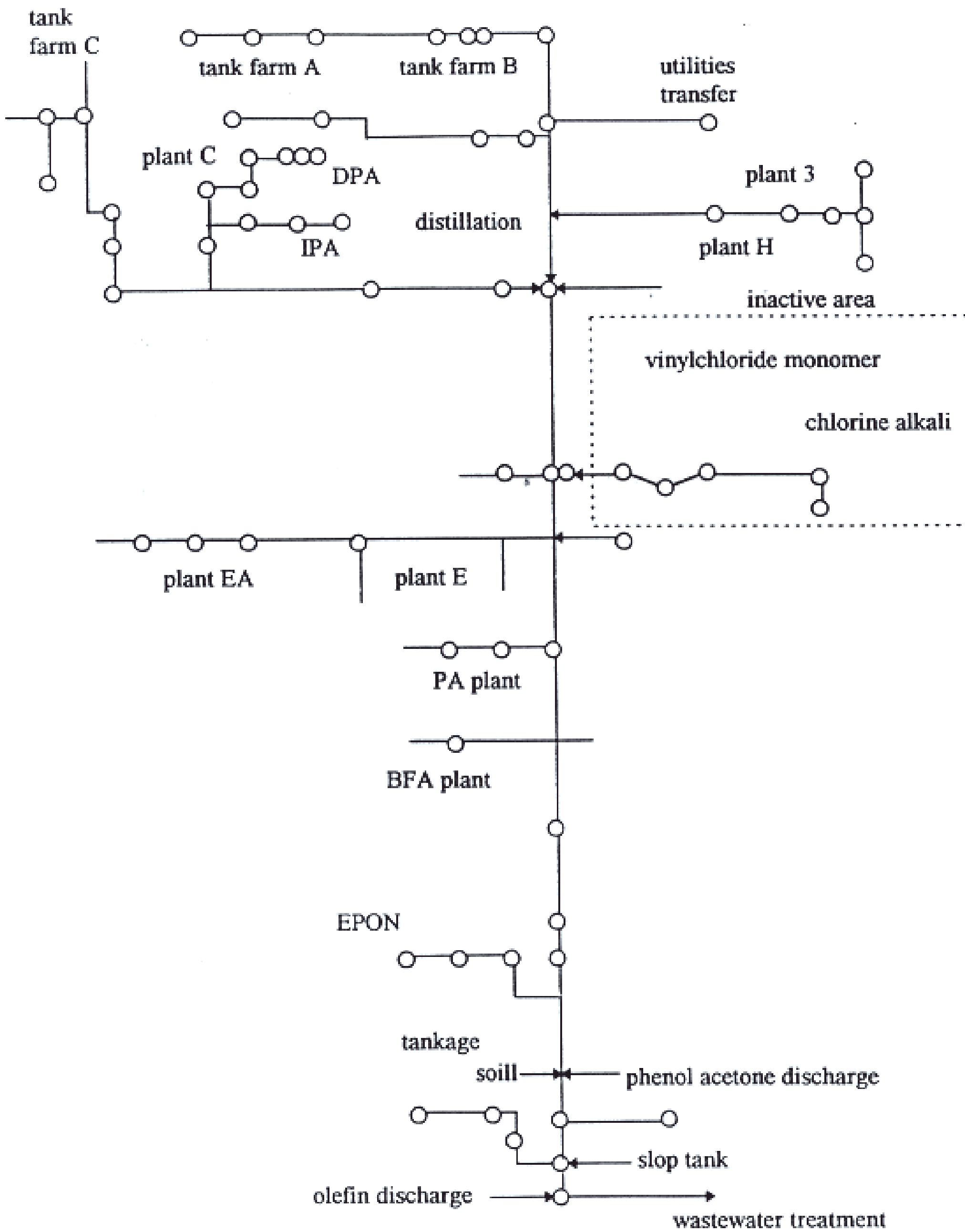


Figure 3.2: Schematic of the OSBL unit used to create GIS files. The unit portrayed is approximately 1500 meters by 500 meters. (Figure not to scale).

3.2 PREPROCESSING

Several steps were taken to prepare data into the necessary format. As the naUTilus/ArcView® tool requires data in a specific format, these steps were part of the procedure followed for each OSBL and ISBL unit. Appendix B includes a detailed description of the data preprocessing commands used in this procedure and Appendix C contains user documentation detailing the procedure, as well as an example for a small sewer network. The following sections give a general description of the procedure.

3.2.1 Digitizing

Most industrial sewer networks tend to be available as hard copy maps or schematics. The ISBL and OSBL data for this study were provided as hard copy schematics. The application of GIS required these networks to be represented in a digitized format. This was accomplished by digitizing the schematics in AutoCad. Each ISBL and OSBL unit was digitized as two data layers: a series of lines (branches) and a series of points (junctions, manholes, and drains). Each line was drawn in the direction of the branch flow. The series of points acted as a reference to ensure that branches were properly digitized. A detailed description of the process used to digitize the sewer networks is included in Appendix B.

Once the schematics were represented in digitized formats, they were processed in Arc/Info. Processing in Arc/Info allowed the connectivity of the branches and nodes to be established. The connectivity was instrumental in the application of naUTilus in ArcView®. This intermediate step created a coverage (map data set) for each ISBL and OSBL unit. The coverages were constructed from the line data layers, with the point data derived from end points of branches. Using the coverages, two themes (map data layers) were created for each ISBL and OSBL unit.

The two themes created for each ISBL and OSBL unit represented the node and line feature of the unit. Each theme was represented by a set of three files called shape files. ISBL node features were described by the files INODES.SHP, INODES.SHX, and INODES.DBF. ISBL line features were described by the files IBRANCH.SHP, IBRANCH.SHX, and IBRANCH.DBF. Similarly, OSBL units were described by the files ONODE.SHP, ONODE.SHX, ONODE.DBF, OBRANCH.SHP, OBRANCH.SHX, and ONODE.DBF. The files INODES.DBF, IBRANCH.DBF, ONODE.DBF and OBRANCH.DBF represent the feature attribute tables (discussed in Section 2.4) for each theme. A detailed description of the commands used in Arc/Info is presented in Appendix B, following the discussion on digitizing.

3.2.2 File Setup

To ensure a smooth connection between the OSBL and ISBL portions of a sewer network, a structure was selected for storing OSBL and ISBL data sets. Not only did this method of storing files simplify the maintainance of data sets, it also allowed the connection between OSBL and ISBL units. Figure 3.5 illustrates the file setup developed for an example network with four ISBL units flowing to the OSBL unit. For further discussion of the file setup, see the user documentation included in Appendix E.

The network described in Figure 3.3 corresponds to the setup used for the OSBL unit. The main directory (sewer1) contained the naUTilus executable files, the OSBL shape files, the ArcView® project file, and three other files that are required by the ArcView® project. The other three files were ArcView® legend files, which stored the color scheme for display of sewer elements. Table 3.1 describes the files listed in Figure 3.3.

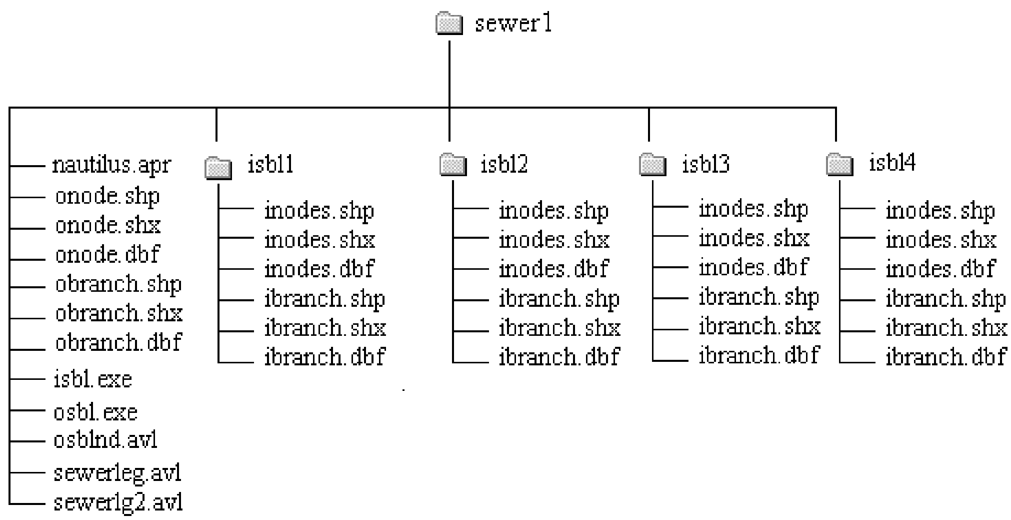


Figure 3.3: File setup for an OSBL unit with four ISBL feed units. The first branch lists the files in the main directory, including the ArcView® project file, the OSBL GIS files (shape files), the naUTilus executable files, and the ArcView® legend files. Two sets of shape files are listed for each ISBL subdirectory.

Table 3.1: Description of files needed for ArcView® connection on initial use. The structure of the files is shown in Figure 3.3.

File Name	**Description**
nautilus.apr	ArcView® project file
onode.shp, onode.shx, onode.dbf	Shape files describing the OSBL sewer elements represented by points (junctions, manholes, ISBL locations)
obranch.shp, obranch.shx, obranch.dbf	Shape files describing OSBL sewer elements represented by lines (sewer reaches)
isbl.exe	Executable file for the naUTilus ISBL module
osbl.exe	Executable file for the naUTilus OSBL module
osblnd.avl	ArcView® legend file used to assign colors to OSBL nodes features
sewerleg.avl	ArcView® legend file used to assign colors to ISBL node features
sewerlg2.avl	ArcView® legend file used to assign colors to ISBL line features
inodes.shp, inodes.shx, inodes.dbf	Shape files describing ISBL sewer elements represented by points (drains, junctions, manholes)
ibranch.shp, ibranch.shx, ibranch.dbf	Shape files describing ISBL sewer elements represented by lines (sewer reaches)

Once the files were set up in the format described by Figure 3.3, pre-processing was complete. All remaining steps were conducted in ArcView®, using programs written and accessed from the ArcView® project. Other files were stored within the structure described in Figure 3.3 and will be described later in this thesis.

3.3 GIS/naUTilus CONNECTION

As described in Section 2.5, several levels of integration are possible between models and GIS. In this research, ArcView® and the naUTilus model were connected using partial integration. The connection between the naUTilus model and ArcView® was established using several scripts (programs written in ArcView®'s object oriented language, Avenue). These scripts allowed data to be entered in ArcView® and stored for each ISBL and OSBL unit. The data were then used to generate the input text files for the ISBL units and the OSBL unit, also done through a script. The naUTilus model was executed for each unit and the results opened and displayed in ArcView®. Figure 3.4 shows a flow chart of the process.

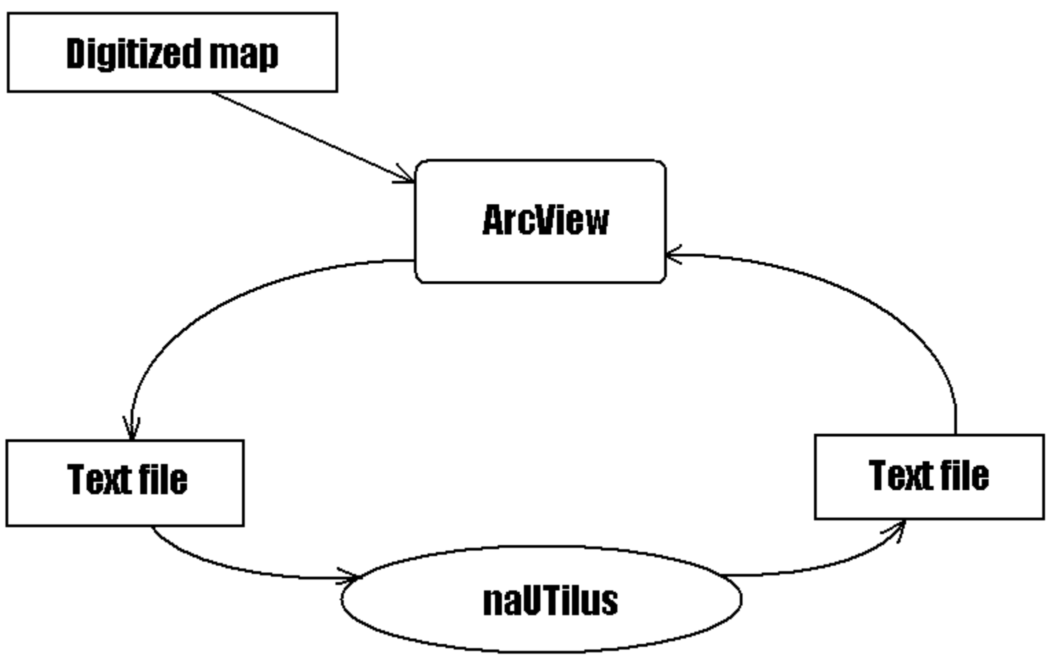


Figure 3.4: Flow chart of the process for the ArcView®/naUTilus connection.

The following sections will describe the programming involved in creating the naUTilus/ArcView® connection, as well as the steps involved in using the naUTilus model in ArcView®.

3.3.1 Programming Involved

All of the programs written for the naUTilus/ArcView® connection were prepared in the object-oriented programming language, Avenue. Programs written in Avenue are called scripts and can be used in several ways: by opening the script and running it directly, by calling it through a menu, by calling it from another script, or by clicking on icons. In this research, scripts are called by clicking on icons or from other scripts activated by icons.

Icons used to call scripts fall into two categories: tools and buttons. A button will execute a script immediately on selection. When a tool is selected it can execute two scripts, one associated with the "click" event and the other associated with the "apply" event. The "click" event executes a script when the tool is first selected. The "apply" event executes a script when a point on the view window is selected. For this research, 33 Avenue scripts were created. The scripts are listed in Table 3.2 with a short description of the function of the script, how it is called, and the type of event that activates it.

Many of the scripts written for the naUTilus/ArcView® connection create, fill, or read information from tables, call tables from a view (map displayed on screen), prompt the user for input, or write output to text files. For ArcView® to work with tables or maps, the files associated with the tables or maps must be part the ArcView® project. The project is the workspace where all tables, maps, and scripts are stored. Initially, the project developed for this research had no tables or maps associated with it. All scripts created to connect naUTilus and ArcView® are stored in the project and have been included in Appendix C.

Figure 3.5 shows the ArcView® window when first opening the naUTilus project, with Button S indicated. Figure 3.6 shows the ArcView® window when working with ISBL and OSBL units. The tool bar and button bar are indicated. Tool 0 through Tool 5 and Tool A through Tool E are visible in the tool bar. Button I, Button N, and Button O are also visible in the button bar.

3.3.2 Theme Initiation in ArcView®

As mentioned in Section 3.3.1, the ArcView® project initially has no themes (data layers) associated with views, and no tables. Using the specified file format discussed in Section 3.2.2, the ArcView® scripts allow tables and views to be created in the ArcView® project to display the ISBL and OSBL units and to store data for each unit. The scripts in the ArcView® project find the shape files for each theme representing the ISBL units and the OSBL units. These shape files are added to the project, with a view opened in the project for each unit.

Figure 3.7 shows the project window in ArcView®. The project window lists all views, tables, charts (graphs created in ArcView®), layouts (displays), and scripts that are part of the ArcView® project when the item is selected in the left portion of the project window. In Figure 3.7, the view item is selected and the list of views in the project is shown. The example shown is for a project with four ISBL units and one OSBL unit.

Table 3.2: Scripts created for the naUTilus/ArcView® connection.

Script Name	Function	Event Type	Activated by
Start	Imports theme files for all ISBL units and OSBL unit. Creates view window for each. Sets legend colors. Start runs *Inittabs*	Click	Button S
Inittabs	Inittabs runs several other scripts and is run once for each ISBL unit.	--	*Start*
Sortpoint	Finds nodes that are numbered in the ArcView® numbering system and not numbered in the naUTilus numbering system for ISBL units.	--	*Inittabs*
Tables	Creates tables for each ISBL unit to hold data describing the ISBL unit	--	*Inittabs*
Drtab	Creates the table for ISBL drains	--	*Inittabs*
Iprompt0	Sets default node type to drain or junction for the ISBL. Run with *Drjunc*	Click	Tool 0
Drjunc	Allows user to change specific nodes from default node type for the ISBL. Run with *Iprompt0*	Apply	Tool 0
Iprompt1	Prompts the user to locate manholes for the ISBL. Run with *Select*	Click	Tool 1
Select	Sets selected nodes as manholes for the ISBL. Run with *Iprompt1*	Apply	Tool 1
Iprompt2	Distinguishes between on-line and elbow drains by location. Sets default drain characteristics. Run with *Draindata*	Click	Tool 2
Draindata	Allows user to change drain characteristics from the default values for individual drains.	Apply	Tool 2
Iprompt3	Prompts user to specify drop locations for the ISBL. Run with *Dropdata*	Click	Tool 3

Table 3.2 (continued): Scripts created for the naUTilus/ArcView® connection.

Script Name	Function	Event Type	Activated by
Dropdata	Allows user to select branches associated with a drop and enter drop characteristics for the ISBL. Run with *Iprompt3*	Apply	Tool 3
Iprompt4	Prompts the user to select locations of hardpipe connections. Run with *Hpselect*	Click	Tool 4
Hpselect	Allows user to specify hardpipe locations and characteristics. Run with *Iprompt4*	Apply	Tool 4
Iprompt5	Sets default branch characteristics and prompts user to select branches to alter from default values for the ISBL. Run with *Selectbr*	Click	Tool 5
Selectbr	Allows user to select branches to edit from default characteristics for the ISBL. Run with *Iprompt5*	Apply	Tool 5
Junction	Sets OSBL nodes at the initial ends of branches as junctions with no manholes, as assumed by naUTilus. Run with *Seljunc*	Click	Tool A
Seljunc	Allows user to alter manhole/junction status of OSBL nodes. Run with *Junction*	Apply	Tool A
Osortpt	Numbers OSBL nodes according to naUtilus numbering scheme.	Click	Tool B
Obrtab	Creates the table describing OSBL branches. Allows user to select default branch characteristics. Run with *Oselbr*	Apply	Tool C
Oselbr	Allows user to edit branch characteristics from the default for individual branches. Run with *Obrtab*	Click	Tool C

Table 3.2 (continued): Scripts created for the naUTilus/ArcView® connection

Script Name	Function	Event Type	Activated by
Odroptab	Creates the table to hold OSBL drop data and prompts the user to select branches associated with drops. Run with *Odropdat*	Apply	Tool D
Odropdat	Allows the user to specify drop locations and characteristics. Run with *Odroptab*	Click	Tool D
Oinflow	Allows the user to specify if inflow to the OSBL exists from non-ISBL sources. Creates a table to hold information an all inflow to the OSBL. Run with *Oflowsel*	Apply	Tool E
Oflowsel	Allows the user to indicate the placement of ISBLs on the OSBL as well as place and specify non-ISBL inflow to the OSBL. Run with *Oinflow*	Click	Tool E
Writeinp	Creates the ISBL input file.	Click	Button I
Isblbat	Creates a batch file to execute ISBL naUTilus and to store files in the correct ISBL subdirectory. Calls *Irunbat* by a delayed run.	Click	Button N
Irunbat	Executes the batch file created in *Isblbat*	--	*Isblbat*
Osblinp	Creates the OSBL input file	Click	Button O
Osblexe	Executes OSBL naUTilus	--	*Osblinp*
Joindata	Joins OSBL naUTilus output to OSBL attribute tables	--	*Osblinp*
Displayemm	Displays OSBL naUTilus output on OSBL view.	--	*Osblinp*

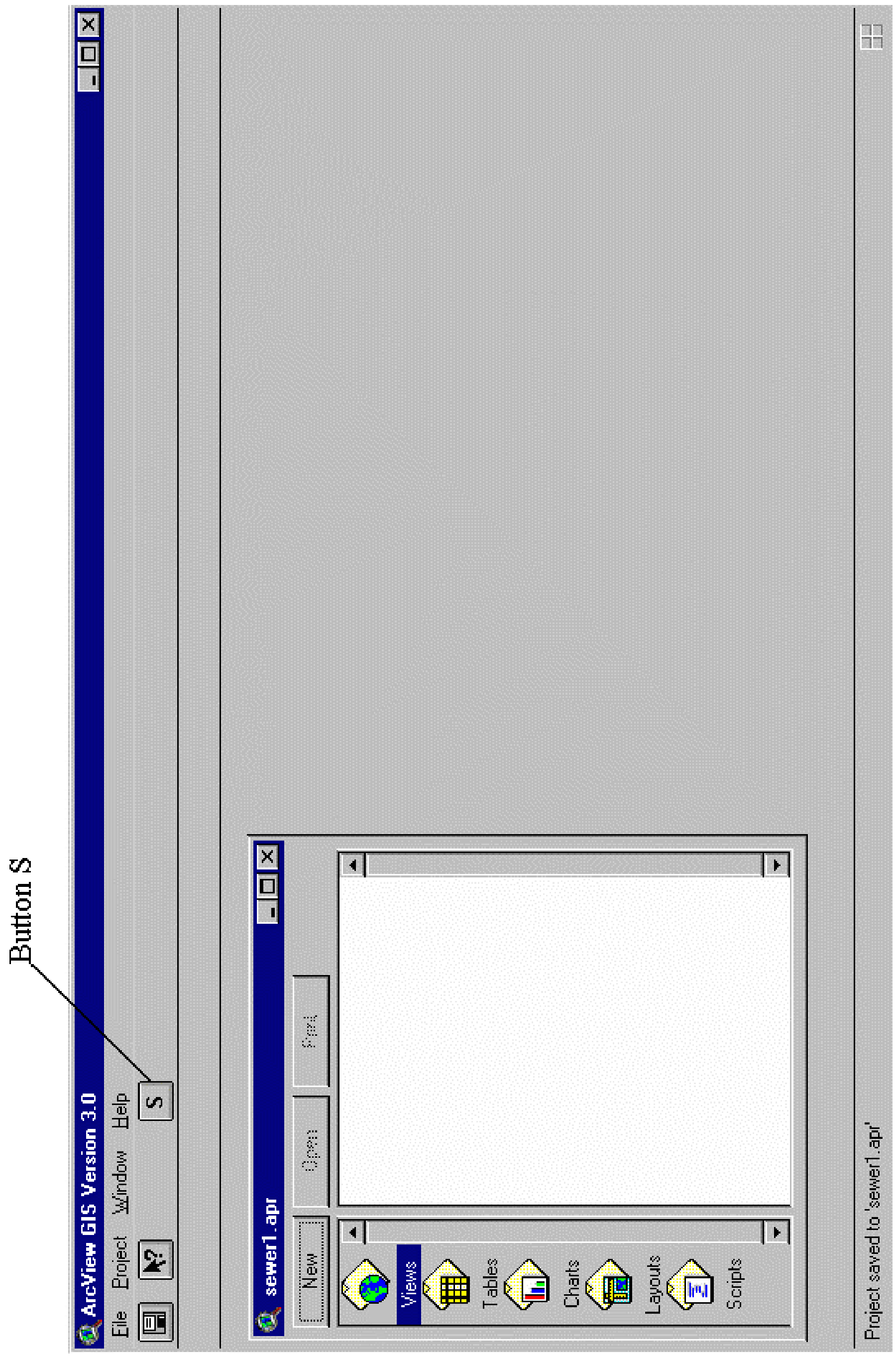


Figure 3.5: ArcView® window that appears when opening the naUTilus project. Button S, which finds and opens the necessary files in the ArcView® project, is indicated.

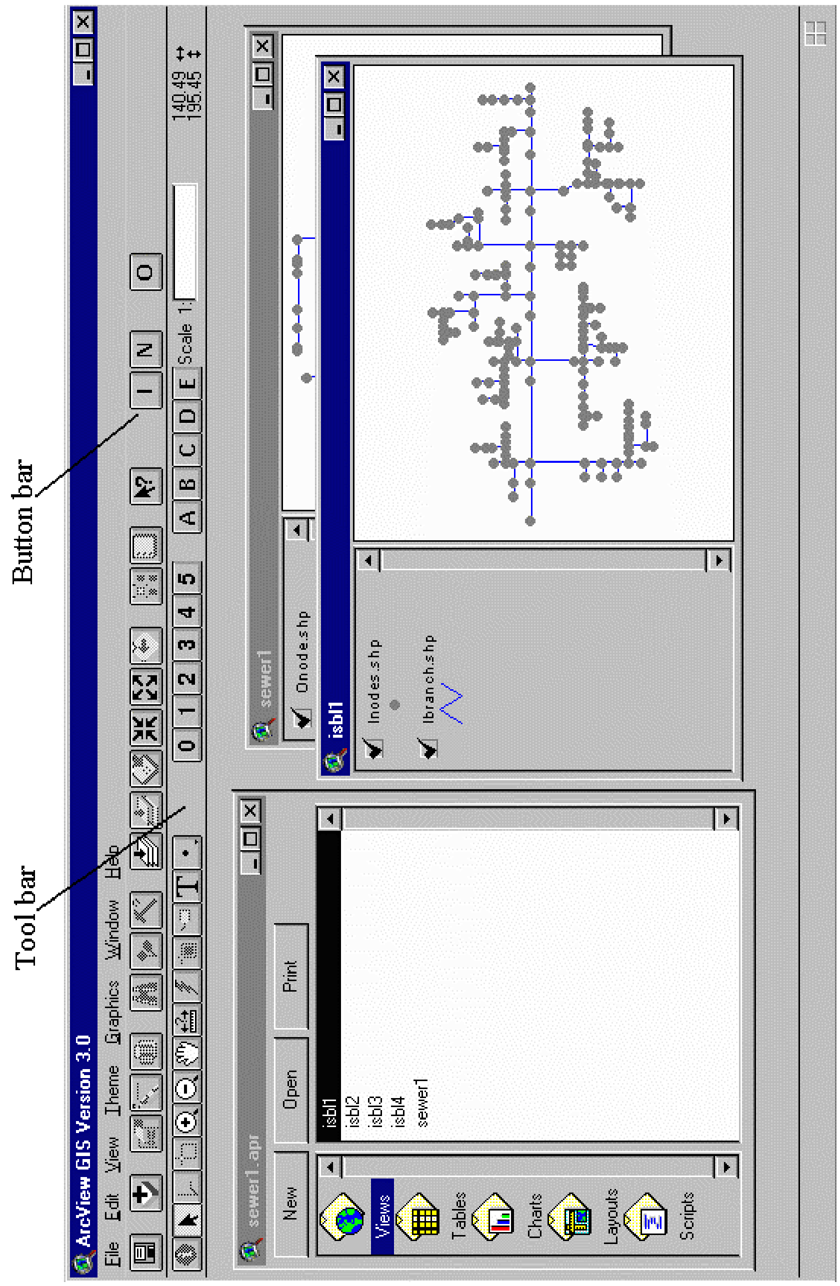


Figure 3.6: ArcView® window visible when working with ISBL or OSBL units. The tool bar and button bar are indicated. Tool 0 through Tool 5 and Tool A through Tool E appear in the tool bar. Button I, Button N, and Button O appear in the button bar.

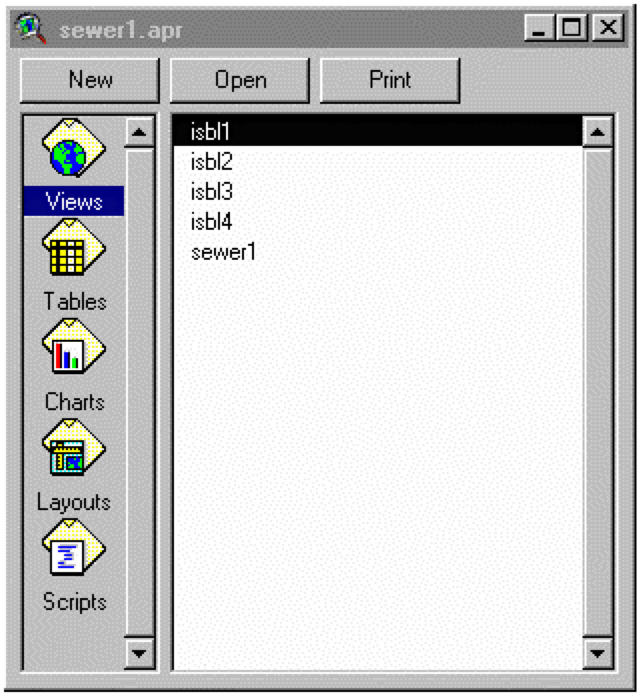


Figure 3.7: Project window for an industrial sewer network with four ISBL units and one OSBL unit (sewer1). The View icon is selected and the views are listed.

3.3.3 Tables Needed for Connection

In addition to initiating the shape files in the ArcView® project, several other files are created for the connection of the naUTilus model and ArcView®. These tables hold the data input by the user describing the sewer network and flow to the network. The tables fall into two categories: ISBL tables and OSBL tables. The units for all values are indicated when the user is prompted for the information, as well as being found in the user documentation included in Appendix E.

3.3.3.1 ISBL Tables

For each ISBL unit, five tables are needed for ISBL data. These tables are in addition to the attribute tables associated with the ISBL themes, INODES.DBF and IBRANCH.DBF. The attribute tables are part of the shape files of each theme, discussed in Section 3.2.1. These additional tables, outlined below, are created and filled through the scripts programmed for the naUTilus/ArcView® connection. All are Dbase files, which can be read in Excel, as well as being opened in ArcView®. Each set of tables is stored in the subdirectory of the appropriate ISBL unit.

Drain Data Table

This table holds information related to all the drains in the ISBL unit. In addition to physical drain characteristics (drain diameter) and flow characteristics (flow rate, concentration, and temperature), information related to the drain type and sealed/unsealed drain status are stored in this table. The drain type (elbow or online drain) is indicated by the value "1" for an elbow drain and a value of "2" for an on-line drain. Sealed drain status is indicated in a similar manner, with a value of "1" for an unsealed drain and a value of "2" for a sealed drain. Figure 3.8 shows part of a drain data table.

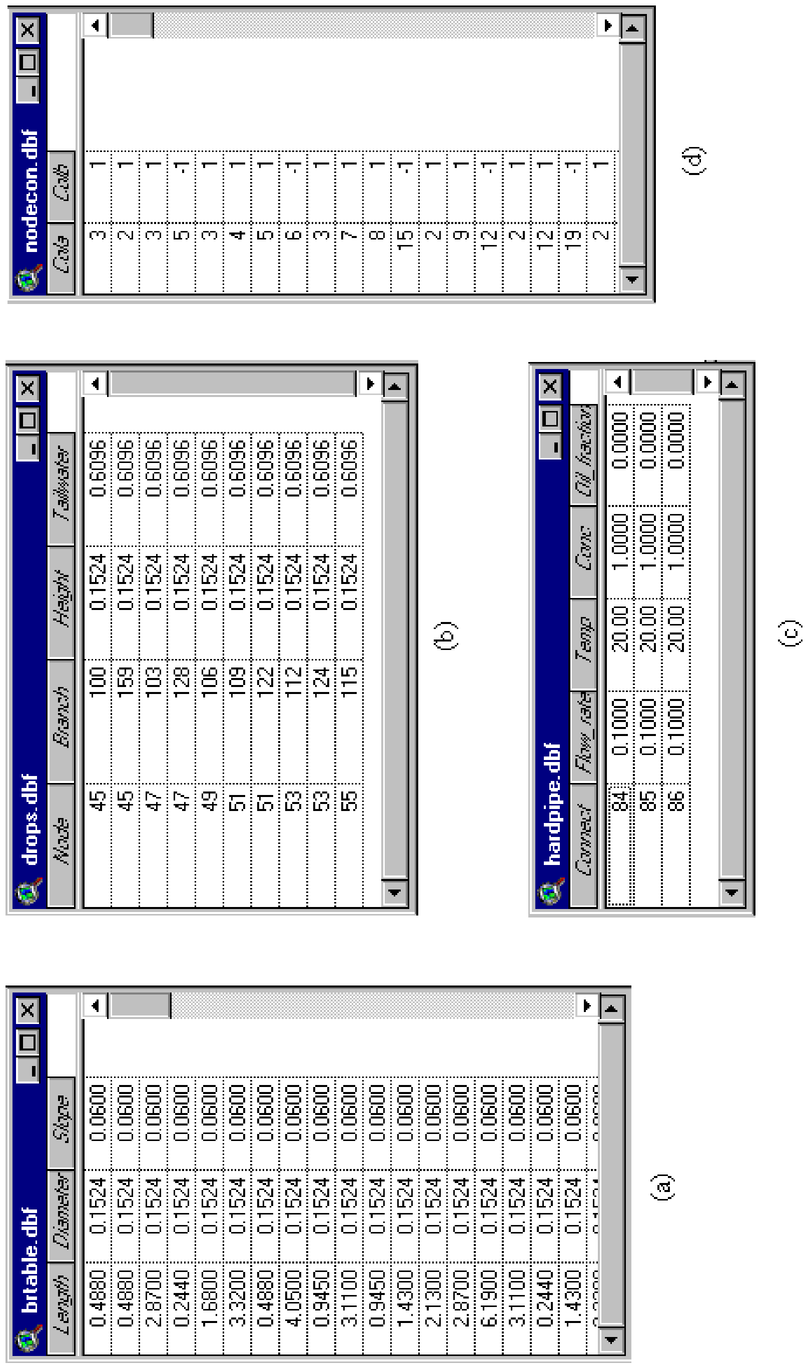
Branch Data Table

The branch drain table stores information on each branch in the ISBL unit. It holds information on the length, diameter and slope of each branch. Figure 3.9 shows an example of a branch data table.

drains.dbf

Isbl_ref	Drain_type	Connectivi	Flow_rate	Temperatur	Concentrat	Diameter	Sealed	Oil_frac
1	1	1	1.0000	30.00	5.0000	0.0508	1	0.0000
2	1	2	0.0000	30.00	0.0000	0.1270	1	0.0000
4	1	4	0.0000	30.00	0.0000	0.2030	1	0.0000
6	1	7	0.0000	30.00	0.0000	0.2030	1	0.0000
9	1	11	0.0000	30.00	0.0000	0.1270	1	0.0000
11	2	6	0.0000	30.00	0.0000	0.1270	1	0.0000
12	1	14	0.0000	30.00	0.0000	0.1270	1	0.0000
14	1	16	0.0000	30.00	0.0000	0.2030	1	0.0000
16	1	21	0.5000	30.00	1.0000	0.1270	1	0.0000
17	1	22	0.0000	30.00	0.0000	0.1270	1	0.0000
18	1	23	1.0000	30.00	0.0000	0.0508	1	0.0000
19	1	24	0.0000	30.00	0.0000	0.1270	1	0.0000
21	1	25	0.0000	30.00	0.0000	0.1270	1	0.0000
22	1	26	0.5000	30.00	0.0000	0.1270	1	0.0000
23	1	27	0.0000	30.00	0.0000	0.2000	1	0.0000
26	1	30	0.0000	30.00	0.0000	0.1270	1	0.0000
27	1	31	0.0000	30.00	0.0000	0.1270	1	0.0000

Figure 3.8: Drain data table. The table holds information describing the drains in an ISBL unit.



(a) brtable.dbf

Length	Diameter	Slope
0.4880	0.1524	0.0600
0.4880	0.1524	0.0600
2.8700	0.1524	0.0600
0.2440	0.1524	0.0600
1.6800	0.1524	0.0600
3.3200	0.1524	0.0600
0.4880	0.1524	0.0600
4.0500	0.1524	0.0600
0.9450	0.1524	0.0600
3.1100	0.1524	0.0600
0.9450	0.1524	0.0600
1.4300	0.1524	0.0600
2.1300	0.1524	0.0600
2.8700	0.1524	0.0600
6.1900	0.1524	0.0600
3.1100	0.1524	0.0600
0.2440	0.1524	0.0600
1.4300	0.1524	0.0600

(b) drops.dbf

Node	Branch	Height	Tailwater
45	100	0.1524	0.6096
45	159	0.1524	0.6096
47	103	0.1524	0.6096
47	128	0.1524	0.6096
49	106	0.1524	0.6096
51	109	0.1524	0.6096
51	122	0.1524	0.6096
53	112	0.1524	0.6096
53	124	0.1524	0.6096
55	115	0.1524	0.6096

(c) hardpipe.dbf

Connect	Flow_rate	Temp	Conc	Oil_fractior
84	0.1000	20.00	1.0000	0.0000
85	0.1000	20.00	1.0000	0.0000
86	0.1000	20.00	1.0000	0.0000

(d) nodecon.dbf

Cola	Colb
3	1
2	1
3	1
5	-1
3	1
4	1
5	1
6	-1
3	1
7	1
8	1
15	-1
2	1
9	1
12	-1
2	1
12	1
19	-1
2	1

Figure 3.9: ISBL tables. Including (a) branch data table, (b) drop data table, (c) hardpipe connection table, and (d) connectivity table.

Drops Data Table

The drops data table describes the height and tail water depth of each drop. The branch and node associated with the drop are indicated in the table. An example of a drops table is shown in Figure 3.9.

Hardpipe Connection Table

The hardpipe connection table holds information on the flow rate, temperature, oil fraction, and concentration of the flow introduced. The location of the hardpipe connection is stored as the node that receives inflow from the hardpipe connection. Figure 3.9 includes an example hardpipe connection table.

Connectivity Table

This table is created using the connectivity established by Arc/Info. It represents the branch/node connectivity of the ISBL system in the fashion required in the naUTilus input file. Figure 3.9 includes an example connectivity table. Connectivity for naUTilus is established by nodes. The first line of input for a node indicates the number of branches connected to that node (both inflowing and outflowing), followed by an indication of whether a manhole exists at the node. The next lines in the table indicate which branches flow into and out of the node, followed by a “1” or “-1”. A “1” indicates that the branch flows into the node and a “-1” indicates an out-flowing branch. The number of lines for each node depends on the value in the first line of input.

For example, the Figure 3.10 shows a schematic of a simple system consisting of two nodes and seven branches. The connectivity table for this system would hold the values shown below the schematic.

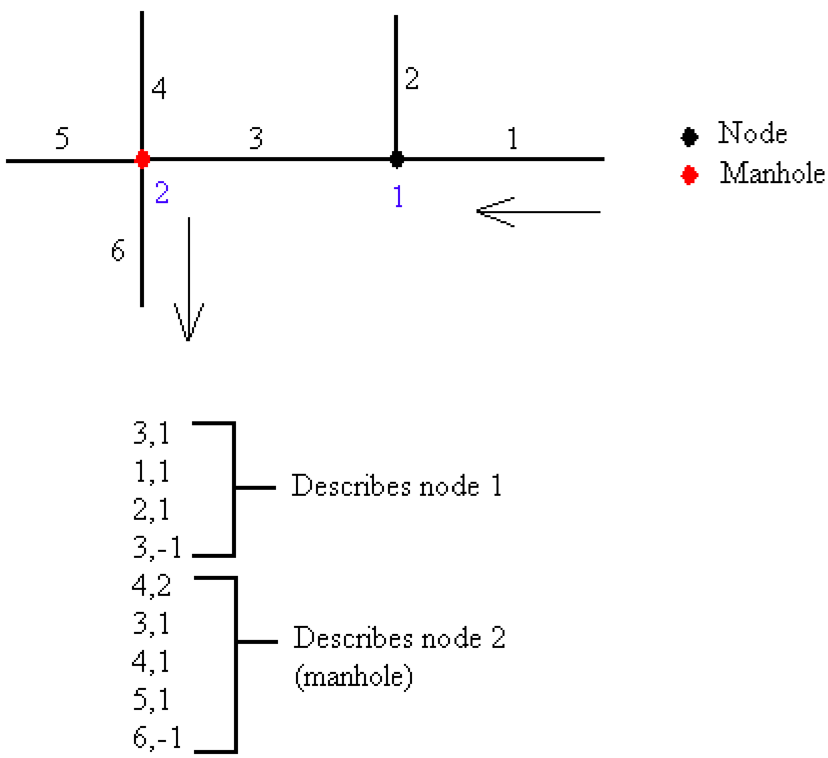


Figure 3.10: Schematic of simple network. Nodes are numbered in blue, branches are numbered in black. The values corresponding to those stored in the ISBL connectivity table are shown below the schematic. The arrows indicate the direction of flow.

3.3.3.2 OSBL Tables

As in the case of ISBL units, several tables are needed for OSBL data, in addition to the attribute tables (ONODE.DBF, OBRANCH.DBF) associated with the OSBL themes. The OSBL tables are created and stored in the main working directory. The tables, created and filled through scripts, are outlined below.

Branch Data Table

The OSBL branch data table holds information on the branch length, slope, and diameter for all branches in the OSBL unit. Each branch has one entry in the branch data table. Additionally, branches 1 meter in length are added to account for locations where manholes exist at initial ends of branches. This was implemented because the naUTilus model does not allow for manholes at inflow points. Adding a short branch is assumed to have no significant effect on predicted emissions. Figure 3.11 illustrates a case where this occurs. An example OSBL branch data table is included in Figure 3.12.

Drop Data Table

The drop data table for the OSBL unit, similar to the ISBL drop data table, holds information on drop height, tail water depth, and drop location. In-flowing branch numbers and the node number at which the drop occurs indicate the drop location. An example of the OSBL drop data table is included in Figure 3.12.

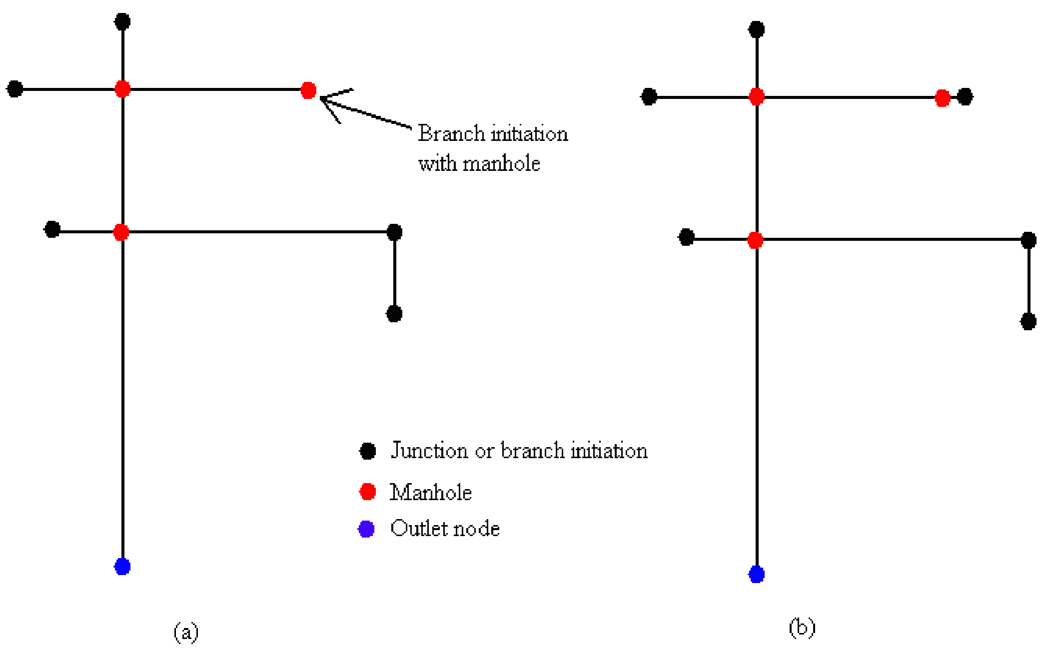
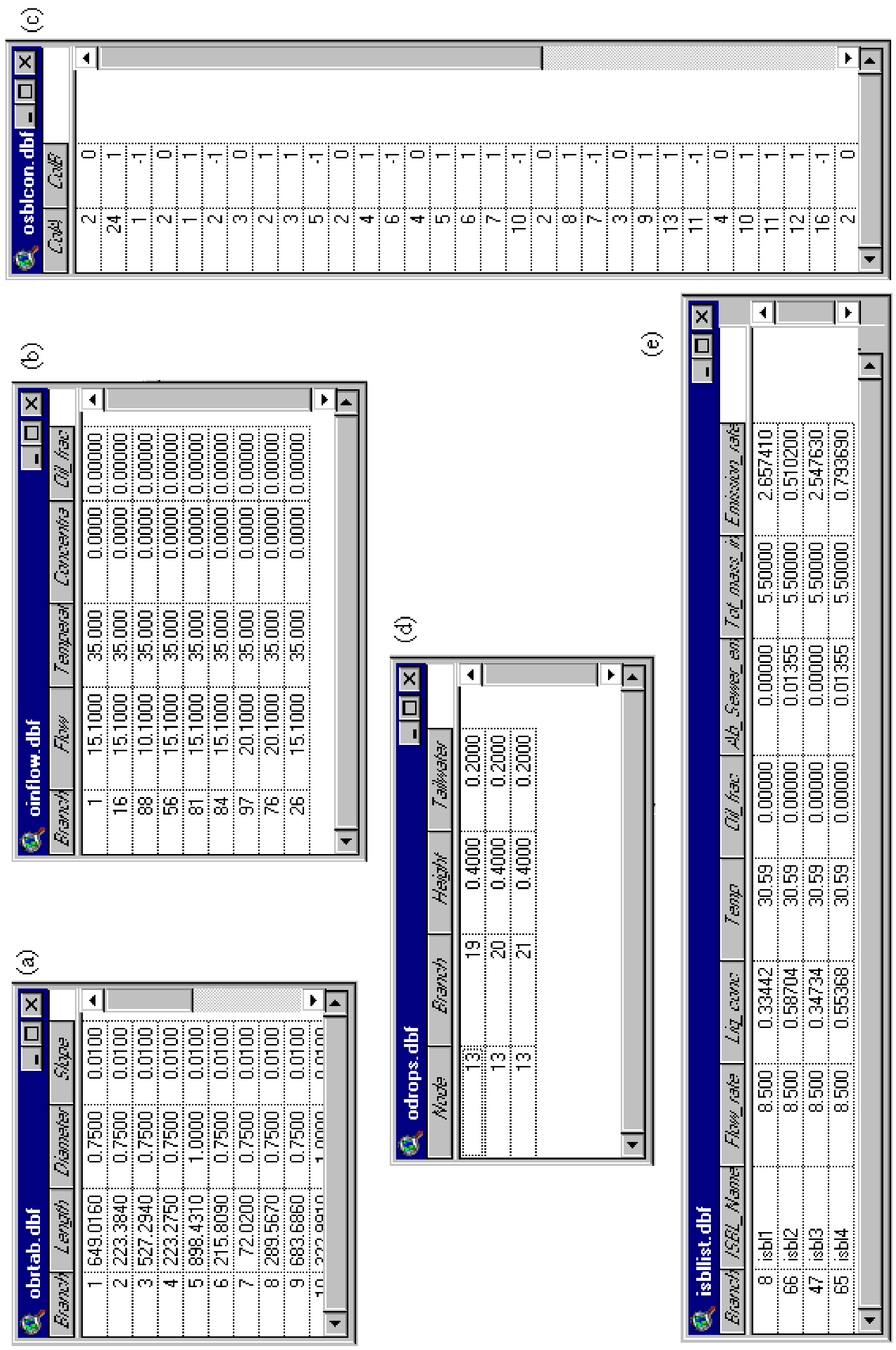


Figure 3.11: A simple OSBL schematic (a) with a manhole present at an initial branch point and (b) altered for naUTilus with additional branch added at manhole. Flow is towards the blue outlet node.



(a) obrtab.dbf

Branch	Length	Diameter	Slope
1	649.0160	0.7500	0.0100
2	223.3840	0.7500	0.0100
3	527.2940	0.7500	0.0100
4	223.2750	0.7500	0.0100
5	898.4310	1.0000	0.0100
6	215.8090	0.7500	0.0100
7	72.0200	0.7500	0.0100
8	289.5670	0.7500	0.0100
9	683.6860	0.7500	0.0100

(b) oinflow.dbf

Branch	Flow	Temperat	Concentra	Oil_frac
1	15.1000	35.000	0.0000	0.00000
16	15.1000	35.000	0.0000	0.00000
88	10.1000	35.000	0.0000	0.00000
56	15.1000	35.000	0.0000	0.00000
81	15.1000	35.000	0.0000	0.00000
84	15.1000	35.000	0.0000	0.00000
97	20.1000	35.000	0.0000	0.00000
76	20.1000	35.000	0.0000	0.00000
26	15.1000	35.000	0.0000	0.00000

(c) osblcon.dbf

ConN	Coeff
2	0
24	1
1	-1
2	0
1	1
2	-1
3	0
2	1
3	1
5	-1
2	0
4	1
6	-1
4	0
5	1
6	1
7	1
10	-1
2	0
8	1
7	-1
3	0
9	1
13	1
11	-1
4	0
10	1
11	1
12	1
16	-1
2	0

(d) odrops.dbf

Node	Branch	Height	Tailwater
13	19	0.4000	0.2000
13	20	0.4000	0.2000
13	21	0.4000	0.2000

(e) isbllist.dbf

Branch	ISBL_Name	Flow_rate	Liq_conc	Temp	Oil_frac	Ab_Sewer_em	Tot_mass_fl	Emission_rate
8	isbl1	8.500	0.33442	30.59	0.00000	0.00000	5.50000	2.657410
66	isbl2	8.500	0.58704	30.59	0.00000	0.01355	5.50000	0.510200
47	isbl3	8.500	0.34734	30.59	0.00000	0.00000	5.50000	2.547630
65	isbl4	8.500	0.55368	30.59	0.00000	0.01355	5.50000	0.793690

Figure 3.12: OSBL tables; (a) branch data table, (b) other inflow table, (c) connectivity table, (d) drop data table, and (e) ISBL table.

Connectivity Table

The connectivity table for the OSBL unit is very similar to the connectivity table for the ISBL unit. It is created through a script, using the connectivity established by Arc/Info. It stores information on the branch/node connectivity of the OSBL unit in the format required by the naUTilus model. For OSBL units, however, manhole locations are indicated through a different method. Consequently, the first line for each node in the OSBL connectivity table consists of one value, indicating the number of branches connected to that node. The OSBL connectivity table is otherwise identical to the ISBL connectivity table discussed in Section 3.3.3.1. An example of a simple OSBL schematic is shown in Figure 3.13. An input file representing the data in the OSBL connectivity table is provided next to the schematic.

Figure 3.12 includes an example of a connectivity table. Note that "0" values in the table indicate that data are being entered for a new node. This corresponds to the lack of a second value on the line describing the number of branches connected to a node, seen in Figure 3.13.

ISBL Table

The ISBL table holds information on the ISBL units that contribute flow to the OSBL unit. One item is created for each ISBL unit in the project. Data held in this table includes ISBL location and summary data describing emissions from the ISBL unit, as well as information on flow to the OSBL generated by the ISBL unit.

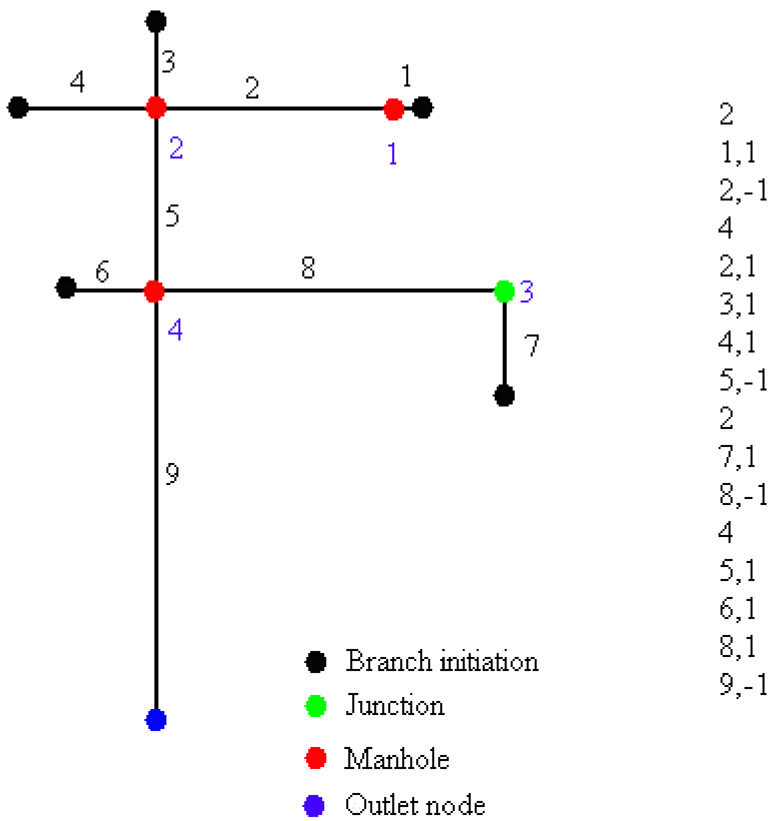


Figure 3.13: Simple OSBL schematic. The values to the right of the schematic show the input required by naUTilus to represent the branch/node connectivity of the OSBL unit.

Summary data include "above sewer" emissions, total mass of chemical entering the ISBL unit, and overall emission rate. Flow information includes flow rate entering the OSBL from the ISBL unit, liquid concentration, liquid temperature, and oil fraction. An example ISBL table is included with the other OSBL tables in Figure 3.12.

Other Inflow Table

A table is also created to hold information on any flows from non-ISBL sources. This may include ISBL units which are not modeled in the ArcView® project but for which flow information is available. Data for other inflows include liquid flow rate, concentration, temperature, and oil fraction, as well as the location of the inflow. Figure 3.12 illustrates a table for other inflow.

3.3.4 Other Files Needed for Connection

In addition to the tables listed, the scripts used to establish the connection between naUTilus and ArcView® created other files. One type of file was the object database (ODB) file. This file stored the names and locations of other files for easy access by the ArcView® project. As each ISBL subdirectory held a version of the tables discussed in Section 3.3.3.1, duplicate file names listed in the ArcView® project would have been cumbersome. Creating an ODB file allowed files of identical nomenclature stored in different subdirectories to be accessed properly. A brief summary of all tables and other files created for the naUTilus/ArcView® connection is included in Table 3.3. This includes the naUTilus input and output files created during the naUTilus/ArcView® connection.

Table 3.3: Summary of tables and files created for the naUTilus/ArcView® connection.

File Name	File Type	Summary/Description
brtab.dbf	dBase	ISBL branch data table
drains.dbf	dBase	ISBL drain data table
hardpipe.dbf	dBase	ISBL hardpipe connection data table
nodecon.dbf	dBase	ISBL connectivity table
drops.dbf	dBase	ISBL drop data table
isbl.odb	ODB	ISBL object data base file storing location of other ISBL files
obrtab.dbf	dBase	OSBL branch data table
odrops.dbf	dBase	OSBL drop data table
osblcon.dbf	dBase	OSBL connectivity table
isbllist.dbf	dBase	Table of ISBL units contributing flow to the OSBL
oinflow.dbf	dBase	Other inflow data table
isbl.in	Text	ISBL naUTilus input file
isbl.out	Text	ISBL naUTilus output file
isblout.txt	Text	ISBL naUTilus output file (comma separated output)
osbl.in	Text	OSBL naUTilus input file
osbl.out	Text	OSBL naUTilus output file
ondout.txt	Text	OSBL naUTilus output file (node data, comma separated)
obrout.txt	Text	OSBL naUTilus output file (branch data, comma separated)

3.3.5 Input File Creation

The tables described in Section 3.3.3 includemuch of the input needed for the naUTilus/ArcView® connection. Other required data include chemical properties, ambient conditions, and calculation options. The user is prompted for these values during input file creation. All other values written to the input files are accessed from the tables discussed in Section 3.3.3 or tables associated with the shape files. Creation of the input file, like creation of and data entry to the discussed tables, is done through scripts written for the naUTilus/ArcView® connection.

Chemical Properties

The naUTilus model requires input on chemical properties for the volatile compound being modeled in the industrial sewer system. Data requirements include the Henry's law constant, the liquid and gas phase diffusion coefficients, and the octanol-water partition coefficient. As discussed in Section 2.3.4, the naUTilus model accepts four methods of entering the Henry's law constant. This data must be provided for both the ISBL and OSBL modules of naUTilus.

Ambient Conditions

Ambient conditions such as wind speed, temperature, and relative humidity are also required by the naUTilus model and are used for estimating air exchange (ventilation) rates between the sewer and ambient atmosphere. These values are required for both the ISBL and OSBL modules of naUTilus.

Calculation Options

As discussed in Section 2.3.4, naUTilus offers options for mass transfer calculations. For ISBL units, the user must specify if the system is kinetics-limited or equilibrium-limited. For both ISBL and OSBL units, the user must specify if mass transfer along reaches will be calculated using relationships from Parkhurst and Pomeroy (1972) or Owens *et al.* (1964). In ISBL units, the user must specify whether mass transfer at drop structures will be calculated using relationships developed by Nakasone (1986) or relationships used in WATER8 (USEPA, 1994).

3.3.6 naUTilus Execution

Execution of the naUTilus model is accomplished from within ArcView® through a script written for the naUTilus/ArcView® connection. For ISBL units, this script creates a batch file. This batch file copies the naUTilus input file for an ISBL unit from the ISBL subdirectory to the main directory where the naUTilus executable files are stored. The batch file then runs the ISBL module of naUTilus, copies the output files to the correct subdirectory, and removes the copies of the files in the main directory. This ensures that all files related to an ISBL are stored in the appropriate directory. As all OSBL files are stored in the main directory, along with the executable naUTilus files, this step is not necessary for the OSBL unit. The OSBL module of naUTilus is run through a script.

Both the ISBL and OSBL naUTilus modules are run through a "delayed run", i.e. a specified period of time passes before the command is passed to the system. This is necessary to avoid any problem that may occur due to memory or system resource requirements. When this error occurs, closing other processes often rectifies the problem.

3.3.7 Opening naUTilus Ouput in ArcView®

Results from execution of the ISBL module of naUTilus are written to the appropriate ISBL subdirectory. Corresponding files are discussed in Section 2.3.1. Examples of naUTilus output are included in Appendix D. Of the ISBL naUTilus output files, only the file ISBLOUT.TXT is used for data in ArcView®. These data become a part of the file ISBLLIST.DBF when ISBL units are placed on the OSBL unit in a process automated by scripts. The other ISBL output file (ISBL.OUT) is available for user reference.

The OSBL module of naUTilus writes three output files, also described in Section 2.3.1. Of these files, two files are opened in ArcView® for data display: ONDOUT.TXT and OBROUT.TXT. The data from these files are joined to the feature attribute tables for the OSBL node and branch themes. Once joined to the feature attribute tables, these data can be displayed on the map of the industrial sewer network. Scripts written for the naUTilus/ArcView® connection join the naUTilus output and attribute tables, and display chemical emissions on the OSBL map.

Chapter 4: Example Application and Results

4.1 ISBL Application and Results

Using the process described in Chapter 3, the ISBL module of naUTilus was applied to the example ISBL unit discussed in Section 3.1. Data related to sewer dimensions for this ISBL unit were provided by the Shell Development Company (Cano, 1997). The sewer dimensions were entered for the ISBL unit. As discussed in Section 3.1, hypothetical values were used for flow to the system.

The ISBL portion of naUTilus was run for three chemicals at varying ambient conditions, liquid flow conditions, and sewer sealed drain conditions. The three chemicals that were examined (methanol, benzene, and 1,3-butadiene) were selected to cover a range of volatilities. The Henry's law constant acts as an indication of the chemical volatility; a high Henry's law constant indicates a highly volatile chemical (1,3-butadiene), while a low Henry's law constant indicates a chemical of low volatility (methanol).

Table 4.1 lists the chemical properties for each of the three chemicals used, as well as the method used to calculate the Henry's law constant. This section describes the conditions for which ISBL naUTilus was run and the results for each case. For each case, one factor was varied from the baseline condition described in Section 4.1.1.

Table 4.1: Chemical property data for the three chemicals examined. Two methods of calculating the Henry's law constant were used. The values of Henry's law constant at 25 ^oC indicate the wide range of volatility the chemicals represent.

	Henry's law constant calculation method	**Henry's law constant at 25^{o}C**[a]	**Constants for calculating Henry's law constant**	**Liquid Phase Diffusion coefficient (cm^2/s)**[d]	**Gas Phase Diffusion coefficient (cm^2/s)**[d]
Benzene	Van't Hoff equation	0.24	A = 5.53 B = 3194[b]	1.0e-5	0.09
1,3-Butadiene	H_c at known T, adjusted by Antoine's equations	2.9	A = 6.85[c] B = 930 C = 239	1.2e-5	0.108
Methanol	H_c at known T, adjusted by Antoine's equations	0.0002	A = 7.9[c] B = 1474 C = 229	1.8e-5	0.164

a. Federal Register. 1994
b. Ashworth, R.A., *et al*. 1988.
c. Dean, J. A., *ed*. 1992.
d. Corsi.. R.L. 1996.

4.1.1 Baseline Conditions

The ISBL portion of naUTilus was executed on a set of baseline conditions. Baseline flow rates and concentrations are described in Figure 4.1. The temperature of all baseline liquid flows was 30 ^oC. The ambient conditions set as baseline were: relative humidity = 50%, temperature = 20 ^oC, and wind speed = 1 m/s. In addition, all process drains were assumed to be "open", e.i., without water seals. The results of running the ISBL module of naUTilus on the baseline conditions are presented in Table 4.2.

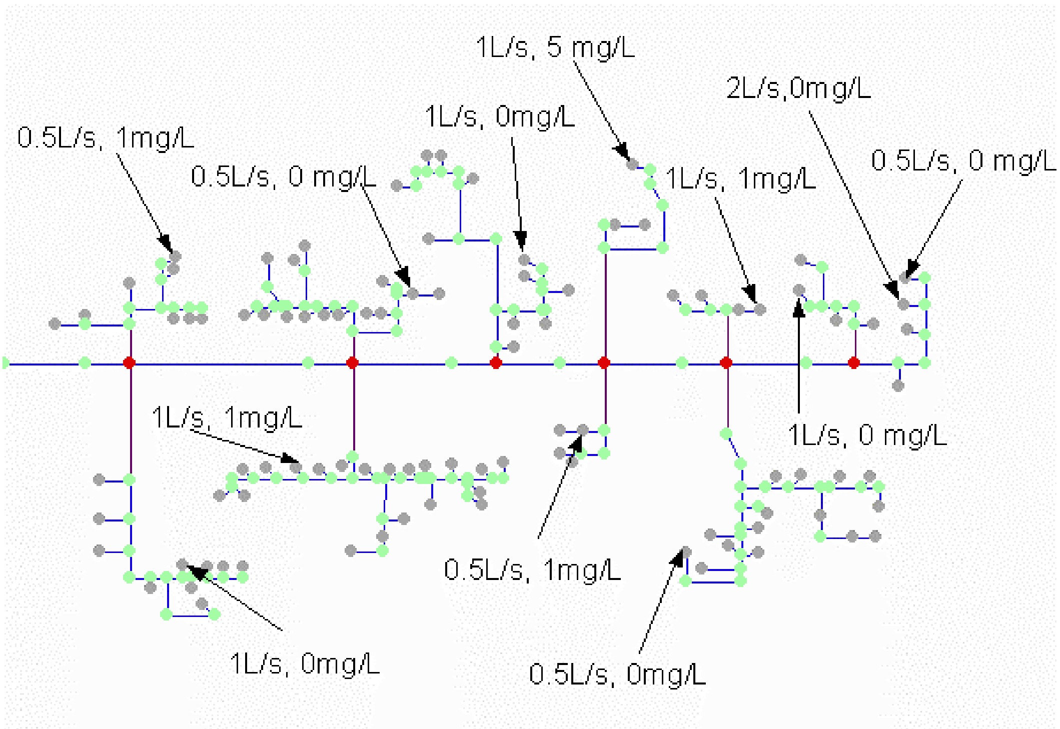


Figure 4.1: ISBL unit with baseline flow rates and concentrations shown with arrows indicating location of the drain with each flow. Gray nodes indicate drains, green nodes indicate junctions, and red nodes indicate manholes.

Table 4.2: Results from naUTilus run on the ISBL unit with hypothetical baseline conditions. Values such as liquid concentration, oil fraction, temperature, and flow rate would be used as input to an OSBL unit.

Chemical	**Above sewer emissions (mg/s)**	**Total mass in (mg)**	**Flow rate (L/s)**	**Liquid conc. (mg/L)**	**Liquid temp. (^oC)**	**Oil frac. (--)**	**Emission rate (mg/s)**	**Stripping efficiency (%)**
Benzene	0.00	8.00	10.50	0.47	30.00	0.00	3.04	38.02
1,3-Butadiene	0.00	8.00	10.50	0.32	30.00	0.00	4.60	57.45
Methanol	0.00	8.00	10.50	0.76	30.00	0.00	0.01	0.09

Emissions from the unit are reported as three values: above sewer emission rate, overall emission rate and stripping efficiency. Above sewer emissions are estimated for ISBL systems with sealed drains. They occur due to mass transfer between the liquid stream and the ambient atmosphere before flow enters a drain. Above sewer emissions from open drains were shown to be negligible in research by Stubbe (1997).

The overall emission rate indicates the mass emission of the chemical from the unit, including above sewer emissions. Stripping efficiency is defined as the total mass of a chemical emitted from the ISBL to the atmosphere divided by the total mass discharged to the unit.

For the baseline conditions, only a small fraction of the methanol was predicted to be removed from the liquid and emitted to the atmosphere. However, nearly 60% of the 1,3-butadiene was predicted to be emitted from the ISBL. This example clearly illustrates the importance of chemical properties on emissions from ISBL units.

4.1.2 Varying Ambient Conditions

Several variations on the baseline conditions were run on the ISBL unit. The effect of ambient wind speed on emissions was examined for three chemicals: benzene, 1,3-butadiene, and methanol. All other environmental factors were held constant, as were flow conditions. Wind speed was changed by altering the ISBL naUTilus input file. Results are presented in Figure 4.2.

Emissions are predicted to increase as the ambient wind speed increased. This increase is most distinct for 1,3-butadiene and benzene, particularly for wind speeds below 1.5 m/s. For both of these chemicals the stripping efficiency approaches a plateau where wind speed no longer led to increased emissions. This phenomenon represents an approach to infinite ventilation ("open system") after a threshold wind speed is achieved. Emissions of methanol, a chemical of low volatility, also increase with increasing wind speed. However, methanol emissions are not significant at any ambient wind speed for the sewer system considered.

The results presented in Figure 4.2 are consistent with the mass transfer principles discussed in Section 2.2.1. As the ambient wind speed increases, the ventilation increases due to increased wind eduction. The higher ventilation rate causes more air exchange with the ambient atmosphere, thereby decreasing the gas phase concentration in the sewer and increasing the concentration driving force. Mass transfer is described by the equation:

$$R_v = K_L A\left(C_l - \frac{C_g}{H_c}\right) \qquad (4\text{-}1)$$

where

R_v = rate of mass transfer from water to adjacent air (mg/s)

K_L = overall mass transfer coefficient (m/s)

A = surface area between liquid and gas (m^2)

C_g = chemical concentration in the gas phase (mg/m^3)

C_l = chemical concentration in the liquid phase (mg/m^3)

As described in Section 2.3.4, the second term in the equation describes the concentration driving force, with C_g/H_c representing a liquid concentration that would be in equilibrium with the existing gas phase concentration.

4.1.3 Varying Liquid Conditions

Two conditions regarding liquid inflow to the ISBL unit were varied: liquid temperature and liquid flow rate. Liquid temperature was varied between 25 ^oC and 50 ^oC and the resulting emissions compared to those from baseline runs, for which the liquid temperature was 30 ^oC. Again, all other flow rates and environmental factors were held constant. Liquid temperature was changed by editing the flow data at each drain where flow existed. Figure 4.3 illustrates stripping efficiencies for various liquid temperatures.

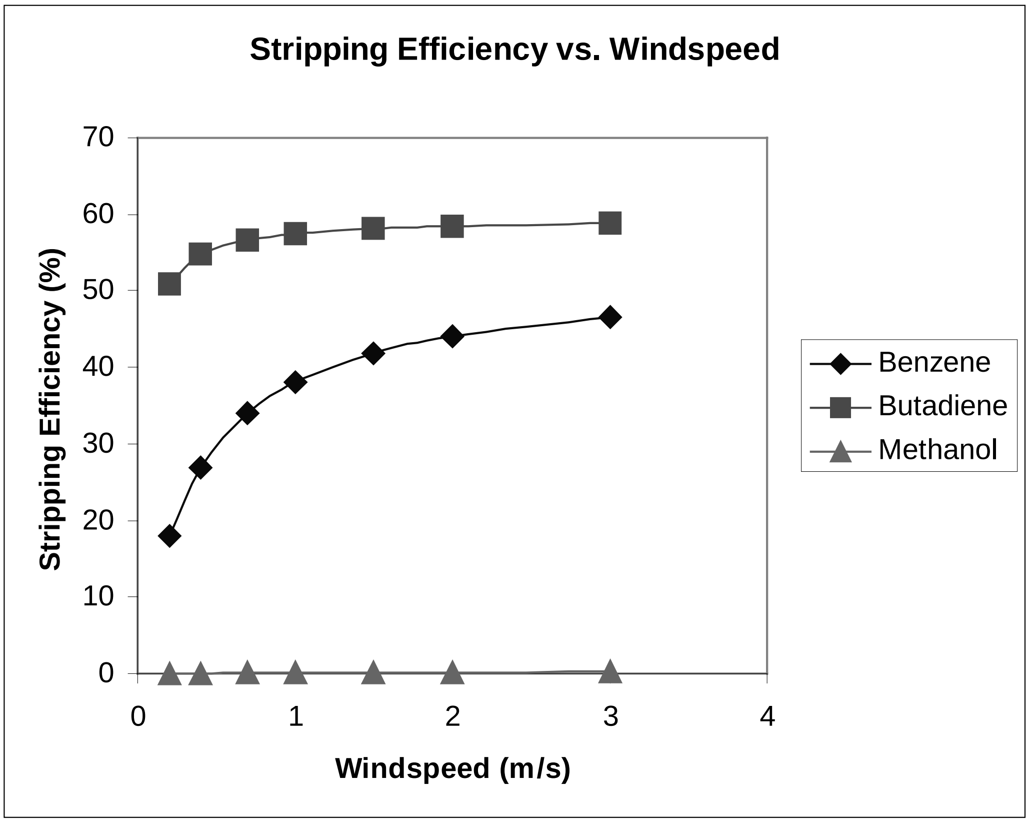


Figure 4.2: Stripping efficiency at various wind speeds. All other environmental and flow characteristics were held constant.

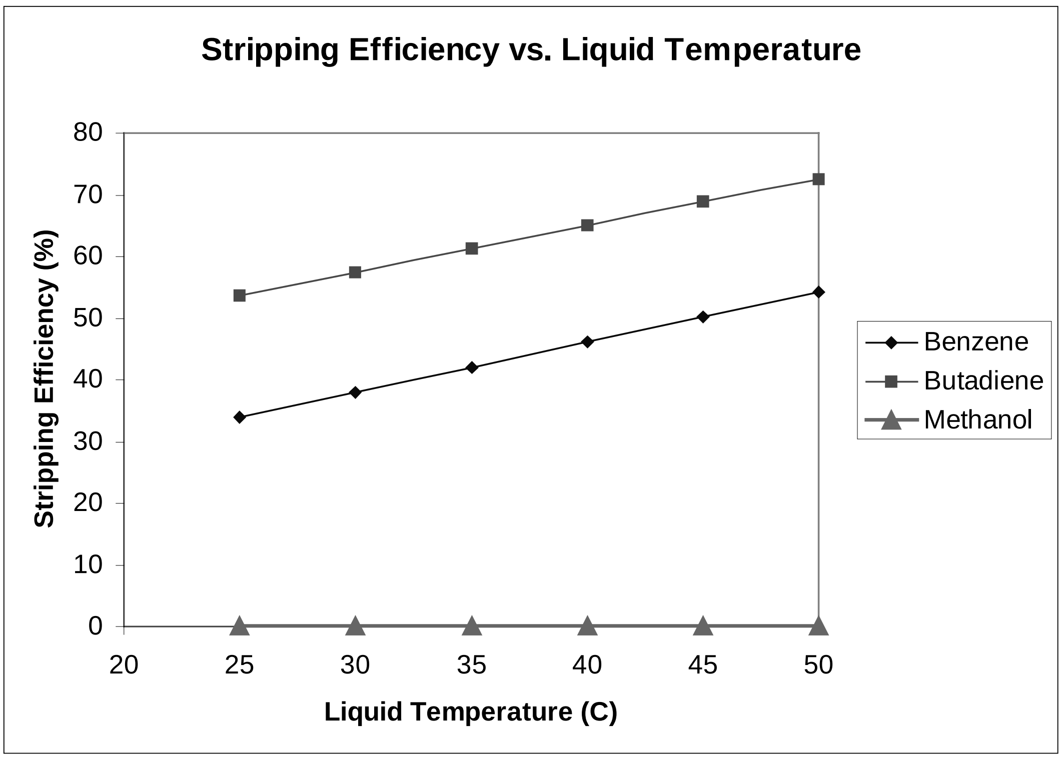


Figure 4.3: Stripping efficiencies for various liquid temperatures, with all other factors held constant.

Emissions of benzene and 1,3-butadiene increase with increasing liquid temperatures. Methanol emissions are predicted to be very low, regardless of the liquid temperature, similar to the case for varying wind speed. The increase in emissions can be attributed to the effect of temperature on Henry's law constant and increased air exchange in the system, both factors being increasing with increasing temperature. The linear nature of the relationship suggests that one mechanism dominates the emissions response to temperature.

Looking at a plot of the variation in Henry's law constant with temperature suggests that the temperature effect on the Henry's law constant is the dominant factor. A plot of the Henry's law constant of benzene over the range of liquid temperatures is shown in Figure 4.4. The plot shows that Henry's law constant increases with increasing temperature, in a relationship that is approximately linear over the range of interest.

The liquid flow rate to the system was varied for the ISBL unit by applying a liquid flow multiplier (LFM). All flows in the system were multiplied by this value.

$$\text{New Liquid Flow} = \text{LFM} \times \text{Baseline Flow}$$

Total flow to the system under baseline conditions was 10.5 L/s. Applying a liquid flow multiplier between 0.25 and 4 resulted in flows between 2.6 L/s and 42 L/s. All other conditions were held constant when varying the liquid flow rate, including chemical concentrations in flows to the system. This resulted in higher mass flow rates to the system and higher mass emissions. However, stripping efficiency was examined as a normalized indication of emissions. Figure 4.5 shows the stripping efficiency for those flow rates.

Mass emissions increase with increasing flow rate, as expected with increasing mass inflow to the system. Figure 4.5, however, indicates stripping efficiency should actually decrease with increasing liquid flow rate. This occurs because the increased flow rate causes a lower hydraulic retention time in the system, decreasing the time available for mass transfer to occur.

4.1.4 Varying Drain Conditions

As discussed in Section 2.1, drains can be sealed or unsealed. The naUTilus/ArcView® application was used to evaluate the effect of emissions from sealed drains. Two factors were examined with regard to sealed drains: number of drains and drain placement. To determine the effect of both factors, various numbers of drains were assumed to be sealed and placed using two patterns: random placement and grouped placement. Examples of sealed drains placed randomly and sealed drains placed by groups are presented in Figure 4.5. Stripping efficiencies resulting from various numbers of sealed drains with random drain placement are presented in Figure 4.6. Percent reductions in emissions are listed in Table 4.3.

As seen in Figure 4.6, stripping efficiency decreases as the number of sealed drains increases. The rate of decrease is affected by the chemical volatility; significant decreases in emissions are not predicted for 1,3-butadiene until approximately 70% of the drains are sealed, while decreases in benzene emissions were predicted for proportions numbers of sealed drains. Emissions of

1,3-butadiene, the most volatile of the chemicals examined, decrease by less than 10% until between 70% and 80% of the drains are sealed. Benzene, a moderately volatile compound, has a 10% decrease in emissions when approximately 30% of the drains are sealed and approximately 40% decrease in emissions with 70 sealed drains (72% of the drains sealed). Methanol also exhibits significant percent reductions of emissions with few sealed drains. However, as methanol is a chemical of low volatility, the mass emissions are very low regardless of the conditions examined.

Emission rates with grouped drain placement were also examined. Drains were placed in clusters determined primarily by subsystems (drains along the same branch off of the main sewer line). Diagrams showing the grouped location of sealed drains are included in Appendix F. Figure 4.7 shows the emission rates and stripping efficiencies for a various number sealed drains placed by groups.

As in the case of randomly sealed drains placement, stripping efficiency decreased as the number of sealed drains increased for grouped seal placement. Similar trends were predicted with grouped seal placement as with random seal placement. Benzene emissions decreased by 50% with 82 sealed drains (84.5% drains sealed) and 69% for 89 sealed drains (91.7% drains sealed) when grouped seal placement was used. With random seal placement, a 63% decrease in emissions was observed for 85 sealed drains (87.6% drains sealed). Table 4.4 shows emission rates and percent reduction in emission rates with the number of sealed drains placed in groups.

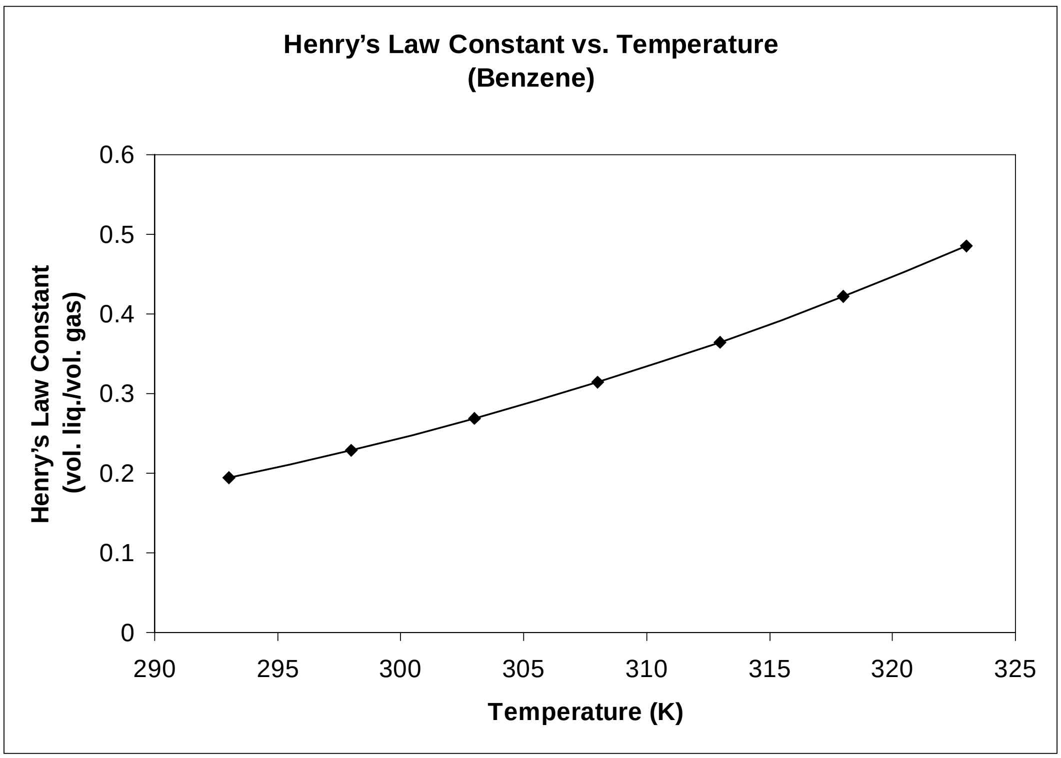


Figure 4.4: A plot of Henry's law constant as affected by temperature.

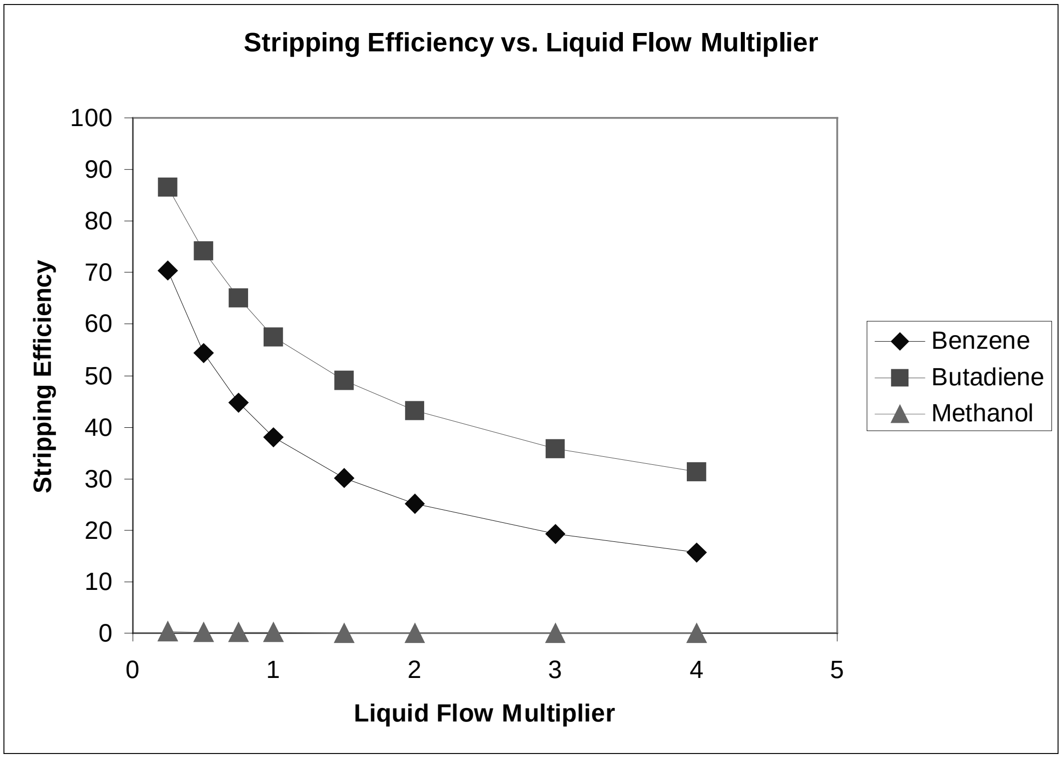


Figure 4.5: Stripping efficiency for various flow rates, as indicated by the liquid flow multiplier.

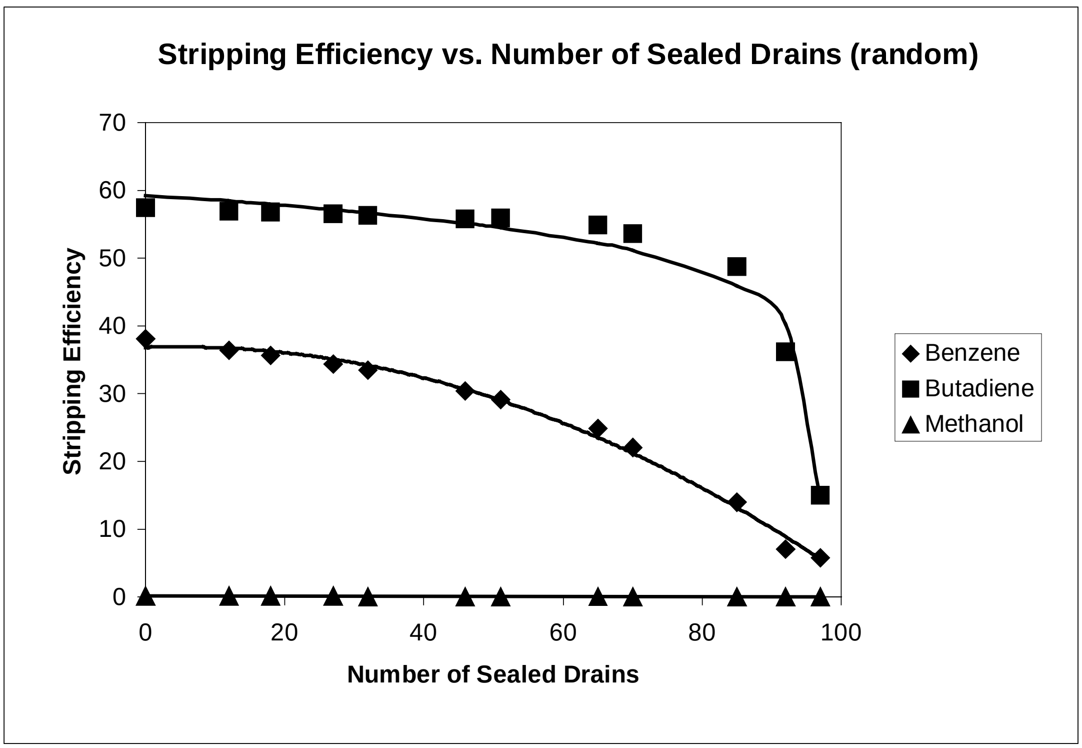


Figure 4.6: Stripping efficiency for various numbers of sealed drains using random placement.

Table 4.3: Emissions and percent decrease in emissions for various numbers of sealed drains, placed randomly. The ISBL has a total of 97 drains.

	Benzene		1,3-Butadiene		Methanol	
Number of Sealed Drains	(mg/s)	% reduction	(mg/s)	% reduction	(mg/s)	% reduction
0	3.04	0.0	4.60	0.0	0.0075	0.0
12	2.91	4.4	4.55	0.9	0.0066	12.2
18	2.85	6.2	4.54	1.2	0.0062	18.3
27	2.75	9.5	4.53	1.5	0.0055	27.5
32	2.68	11.9	4.50	2.1	0.0051	32.7
46	2.43	20.2	4.47	2.8	0.0040	46.9
51	2.33	23.4	4.47	2.8	0.0036	51.7
65	1.99	34.6	4.39	4.6	0.0029	62.0
70	1.77	41.9	4.28	6.8	0.0022	70.9
85	1.12	63.1	3.90	15.2	0.0013	82.3
92	0.56	81.5	2.89	37.2	0.0008	89.5
97	0.46	84.9	1.20	73.9	0.0005	93.1

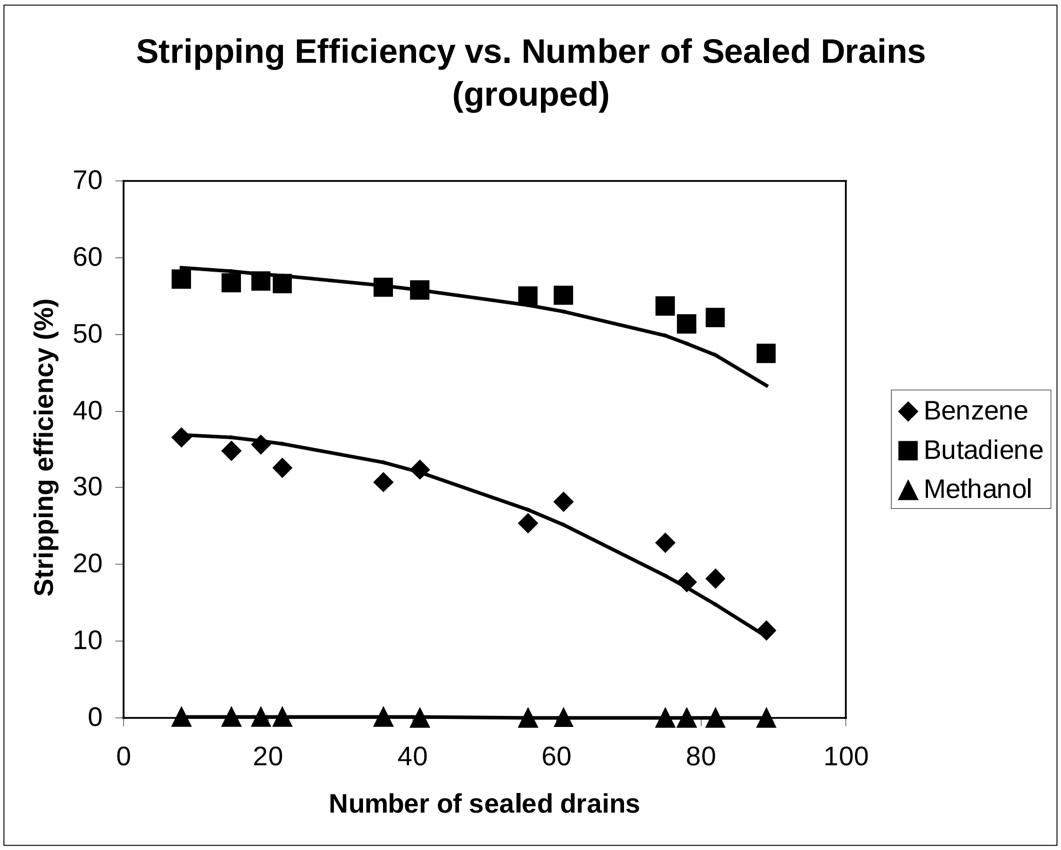


Figure 4.7: Stripping efficiency for various numbers of sealed drains, placed by groups. The lines show the trends corresponding to random placement of sealed drains

Table 4.4: Emissions and percent reduction in emissions for various numbers of sealed drains, with grouped placement.

	Benzene		1-3, Butadiene		Methanol	
Number of Sealed Drains	(mg/s)	% reduction	(mg/s)	% reduction	(mg/s)	% reduction
8	2.92	0.0	4.57	0.0	0.007	0.0
15	2.78	4.7	4.54	0.7	0.006	8.0
19	2.85	2.5	4.56	0.3	0.006	12.0
22	2.61	10.7	4.53	1.0	0.006	11.0
36	2.46	15.8	4.49	1.7	0.005	26.6
41	2.59	11.4	4.46	2.4	0.004	36.3
56	2.03	30.5	4.40	3.8	0.004	48.8
61	2.25	23.0	4.40	3.7	0.003	58.8
75	1.83	37.3	4.29	6.1	0.002	74.1
78	1.41	51.7	4.11	10.2	0.002	72.9
82	1.45	50.2	4.17	8.7	0.002	77.3
89	0.91	68.8	3.80	16.9	0.001	85.1

When comparing Figure 4.6 and Figure 4.7, more scatter was predicted in the graph representing stripping efficiency versus number of sealed drains for grouped placement. This scatter, most evident for benzene, indicated that seal placement did affect stripping efficiency. However, variation was relatively small, implying the effect of seal drain placement was small compared to the effect of increasing the number of sealed drains.

4.2 OSBL Application Results

As mentioned in Section 3.1, the OSBL examined in this research was assigned typical values for its sewer dimensions. It was also assigned hypothetical flow values. Four ISBL units, based on the real ISBL system discussed in this research with various sealed drain conditions, were associated with the OSBL unit. These four ISBL units, with hypothetical flow data, generated the inflow data used to demonstrate OSBL naUTilus. Figure 4.8 shows the locations of the four ISBL units assigned to the OSBL unit, as well as the other flows assigned to the unit. Node types (node with no manhole, node with a manhole, and junction with no manhole) are also shown in Figure 4.8.

Figure 4.9 shows the ISBL unit with the hypothetical flow rate, concentration, and liquid temperatures. The arrows indicate the nodes where the flow is located. These flow conditions were used for each of the four ISBL units. The only variable was the sealed drain status of the hypothetical ISBLs. The sealed drain placement for the ISBL units are described in Appendix F. ISBL1 had all drains unsealed. ISBL2 had all drains sealed. ISBL3 and ISBL4 had a mixture of sealed and unsealed drains; each drain sealed in ISBL3 is unsealed in ISBL4 and each drain unsealed in ISBL3 is sealed in ISBL4.

Benzene was the chemical modeled for the hypothetical ISBL units and for the OSBL unit. An ambient temperature and wind speed of 20 ^{o}C and 1 m/s were used, respectively, with a relative humidity of 50%. The chemical properties used for modeling benzene throughout the sewer were identical to those used to model benzene in the ISBL unit, as listed in Table 4.1. The total mass entering the system was assumed to be 15.5 mg/s.

Typical values for branch diameters and slopes were assigned the to OSBL unit. The branch diameter for main lines was set at 1 meter. Other branches were assigned a diameter of 0.5 meter. Figure 4.10 shows the OSBL system with the main branches in green. No drops were assumed in the OSBL system.

Results from running the OSBL module of naUTilus on the system described above are shown in Figure 4.11 as seen in ArcView®. Figure 4.11 also shows a table summarizing the data on the four hypothetical ISBL units that discharge to the OSBL unit. The results in Figure 4.11 were displayed as colored nodes, as described in the legend in the left-hand portion of the figure. Emissions from the OSBL unit were divided into four equal ranges, determined by the minimum and maximum values output by the naUTilus. Yellow represented nodes with low emissions (0 - 0.162 mg/s), orange and dark orange represented middle range emissions (0.162 - 0.324 mg/s and 0.324 - 0.485 mg/s), and red represented nodes with the highest emission rates (0.485 - 0.647 mg/s). The emissions ranged from 0 mg/s to 0.65 mg/s, with total mass emissions of 5.44 mg/s and a stripping efficiency of 35.11%. An output file of the results (OSBL.OUT) is included in Appendix D.

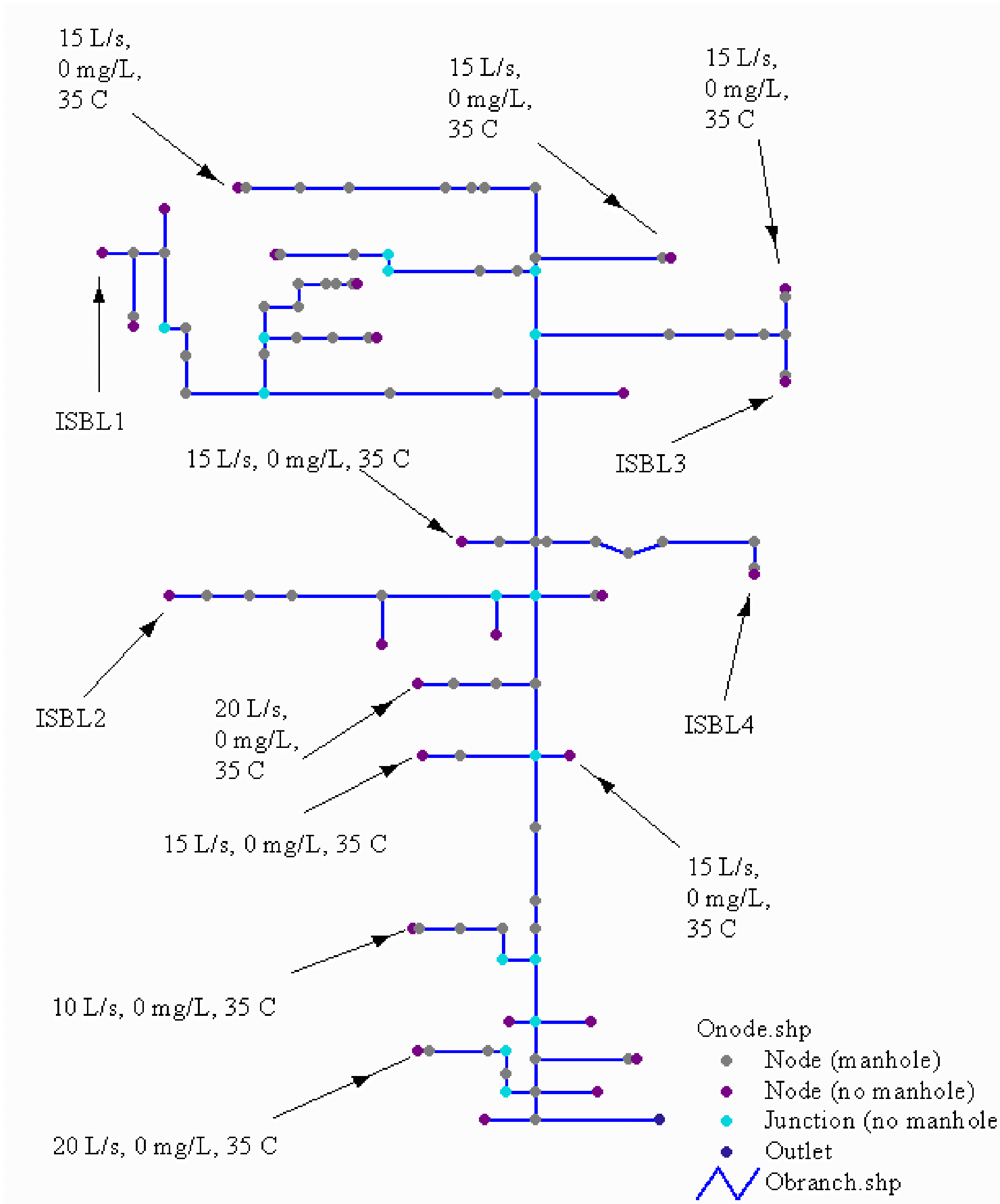


Figure 4.8: Summary of all other inflows to the OSBL system.

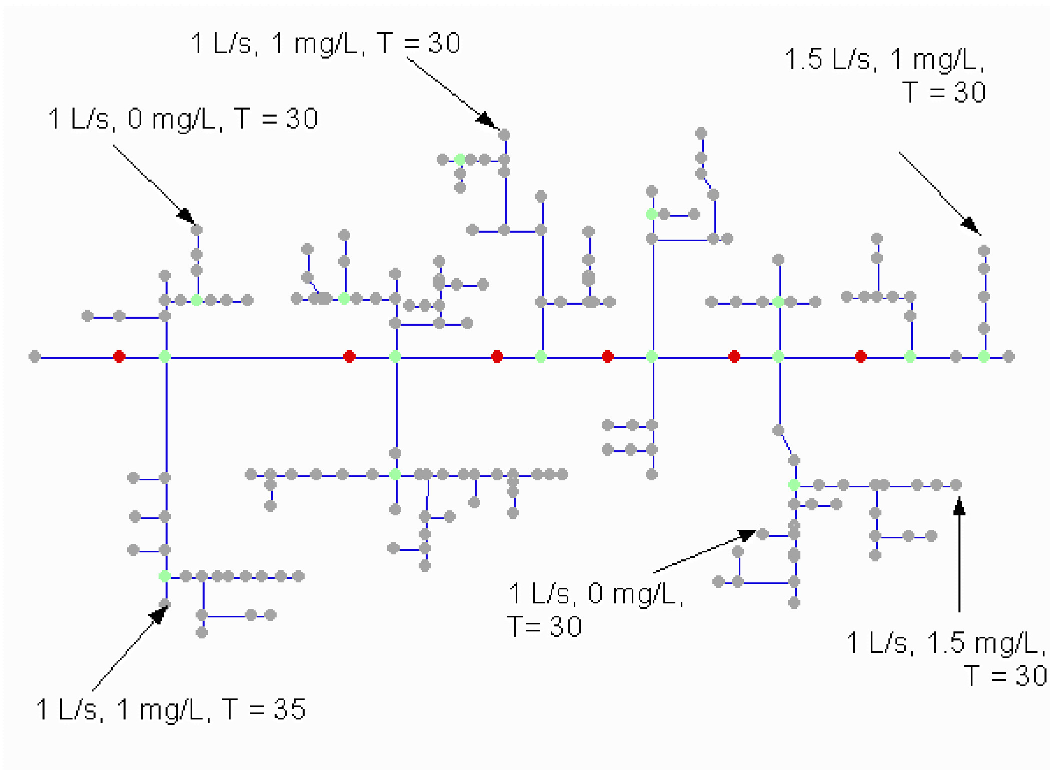


Figure 4.9: ISBL unit with flow indicated for active drains. The flows shown were used in all ISBL units from which OSBL flow was generated. Green nodes indicate junctions, gray nodes indicate drains, and red nodes indicate manholes.

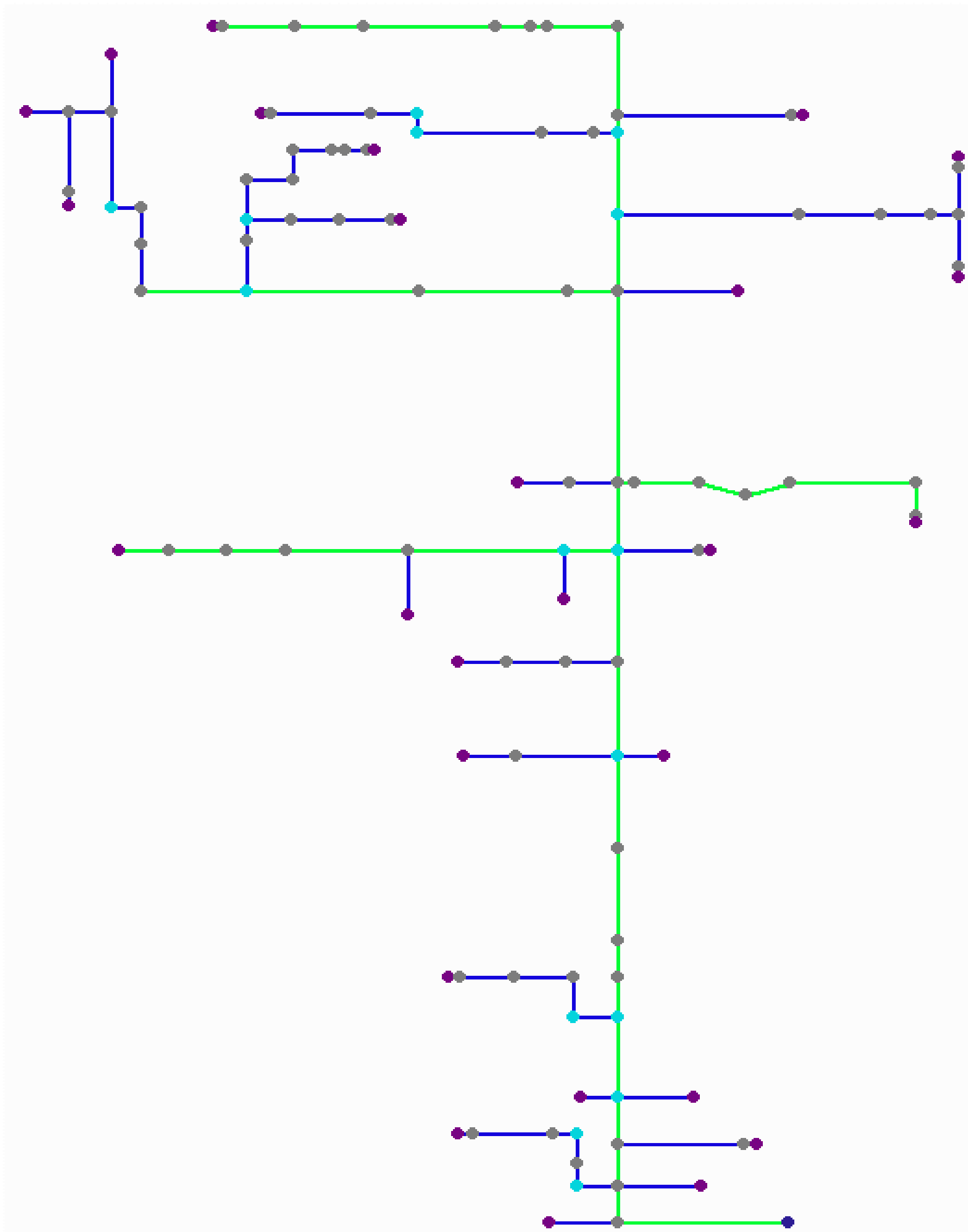
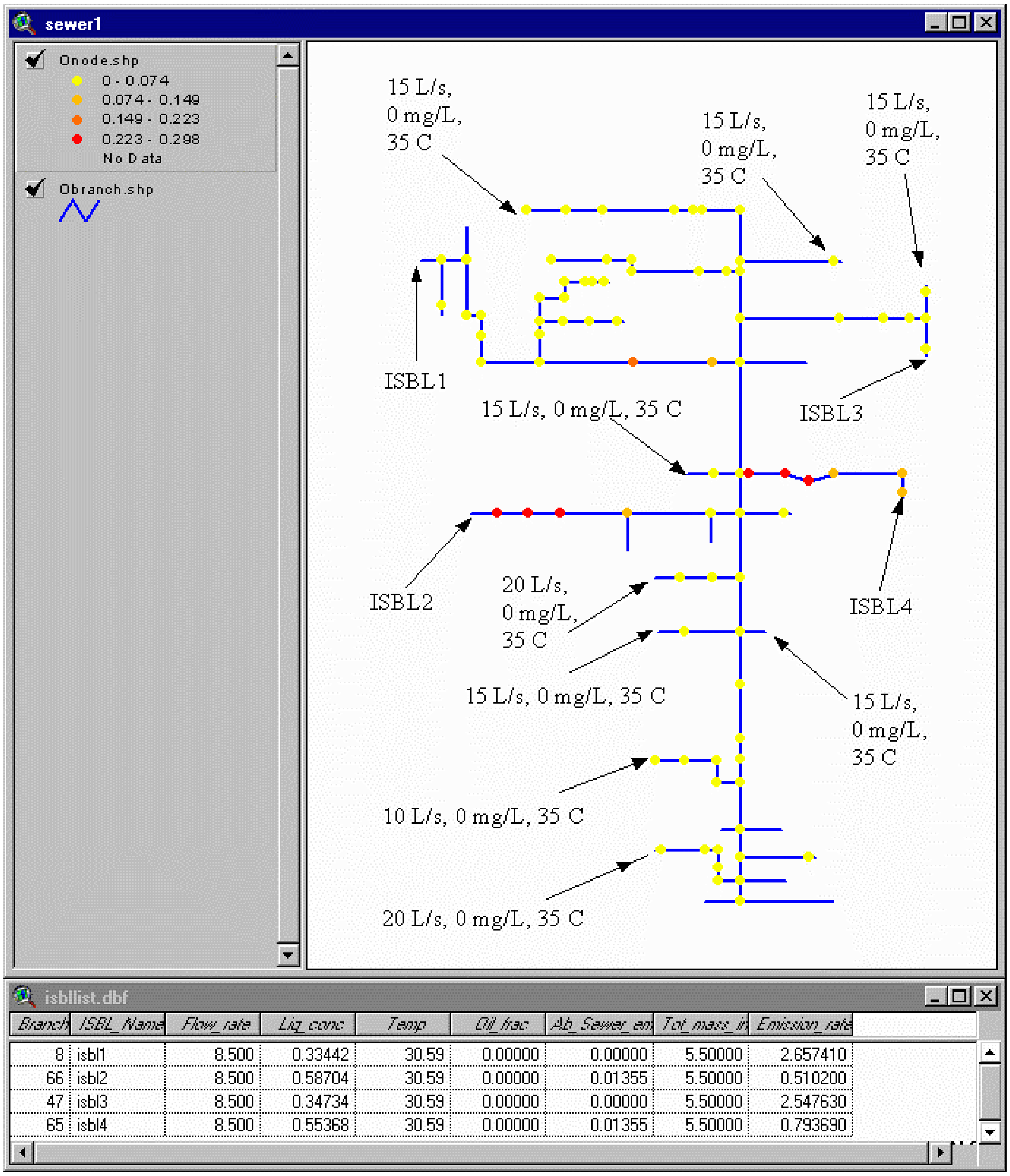


Figure 4.10: OSBL unit with branches assigned a diameter of 1 meter shown in green.

The results in Figure 4.11 also showed the utility of GIS in identifying emissions hot spots. As expected, higher ranges of emissions due to inflow from the unit ISBL2 occurred at the openings immediately following the point of discharge. The mass entering the system was stripped out and emitted to the atmosphere through the first openings downstream of the point of discharge. The effect on emissions due to mass entering the system from ISBL1 and ISBL4, however, did not follow the expected trend. Higher emissions occur at points further downstream from the point of discharge.

The deviation from the trend can be explained by examining the ventilation rate. Figure 4.12 shows the ventilation rate for each node downstream from ISBL1, and indicates that the ventilation rate from the nodes directly downstream from ISBL1 were low (0 to 0.556 L/s), while the nodes further downstream had a ventilation rate of 1.970 L/s. Figure 4.11 shows higher mass emissions of the chemical at the nodes further downstream, corresponding to the nodes with a higher ventilation rate. The higher rate of gas exchange was responsible for the higher emissions.

The higher gas exchange rates were possibly caused by numerous factors, including different hydrodynamics in the branches and the algorithm used by naUTilus in calculating air exchange rates. As shown in Figure 4.10, the branches further downstream from ISBL1 have larger diameters than those immediately downstream from the ISBL unit. This change in diameter possibly affected the hydrodynamics and thus affected mass transfer in the reaches.



Branch	ISBL_Name	Flow_rate	Liq_conc	Temp	Oil_frac	Ab_Sewer_em	Tot_mass_i	Emission_rate
8	isbl1	8.500	0.33442	30.59	0.00000	0.00000	5.50000	2.657410
66	isbl2	8.500	0.58704	30.59	0.00000	0.01355	5.50000	0.510200
47	isbl3	8.500	0.34734	30.59	0.00000	0.00000	5.50000	2.547630
65	isbl4	8.500	0.55368	30.59	0.00000	0.01355	5.50000	0.793690

Figure 4.11: OSBL results with emissions values colored by ranges. The four ranges were determined by the maximum and minimum values output from naUTilus. They are listed in mg/s. The nodes where emissions are highest appear in red. Nodes with no emissions or low emissions appear in yellow. Also shown is a table summarizing the ISBL inflows to the OSBL unit.

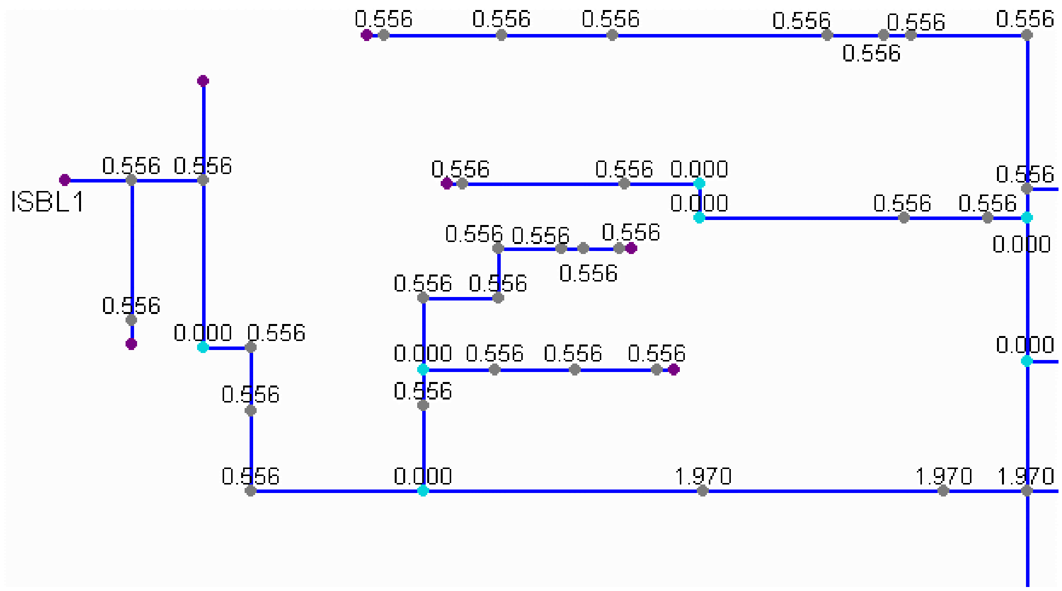


Figure 4.12: Portion of the OSBL unit including ISBL1 and the nodes directly downstream from ISBL1. The nodes are labeled by the ventilation rate in L/s. Purple nodes indicate initial nodes with no manholes, gray nodes indicate manhole locations, and light blue nodes indicate internal nodes with no manholes (junctions). Junctions have a ventilation rate of 0.

Another factor that could have had a large effect on ventilation was associated with the algorithm used by naUTilus to calculate ventilation rate due to water drag. The algorithm for this mechanism assumed that 45% of the nodes were ingassing and 55% of the nodes were outgassing. Those nodes furthest from the outlet of the system were considered ingassing and the other nodes were considered outgassing.

In the case of those nodes directly downstream of ISBL4, the associated branches had identical characteristics, i.e. slope and diameter values. Thus, hydrodynamics were not likely the determining factor in the location of emissions. It was likely that the ventilation algorithm dominated the location of emissions.

The apparent dominance of the algorithm on which node exhibits high emissions brought question to the validity of examining the location of emissions using naUTilus. This determination, however, would have been difficult without the ability to view the ventilation rates and other factors on the visual display of the OSBL network.

Examining some of the other results from naUTilus, little could be determined about emissions resulting from the mass contribution of ISBL3 with the ranges shown in Figure 4.13. To gain a better understanding of low end emissions (emission rates ranging from 0.0 mg/s to 0.162 mg/s), Figure 4.13 shows the same OSBL unit and results with more ranges displayed. Unlike the ranges in Figure 4.11, these ranges were not equally divided through the range of emission values. The ranges were skewed to better represent the low range values. Two additional ranges were added to the existing ranges and the range from 0.0 - 0.162 mg/s was divided into three new ranges. Under the new coloring scheme, yellow represented nodes with no emissions. Two other ranges, 0.001-0.08 mg/s and 0.08 - 0.162 mg/s were defined by colors between yellow and orange, as seen in the Figure 4.13.

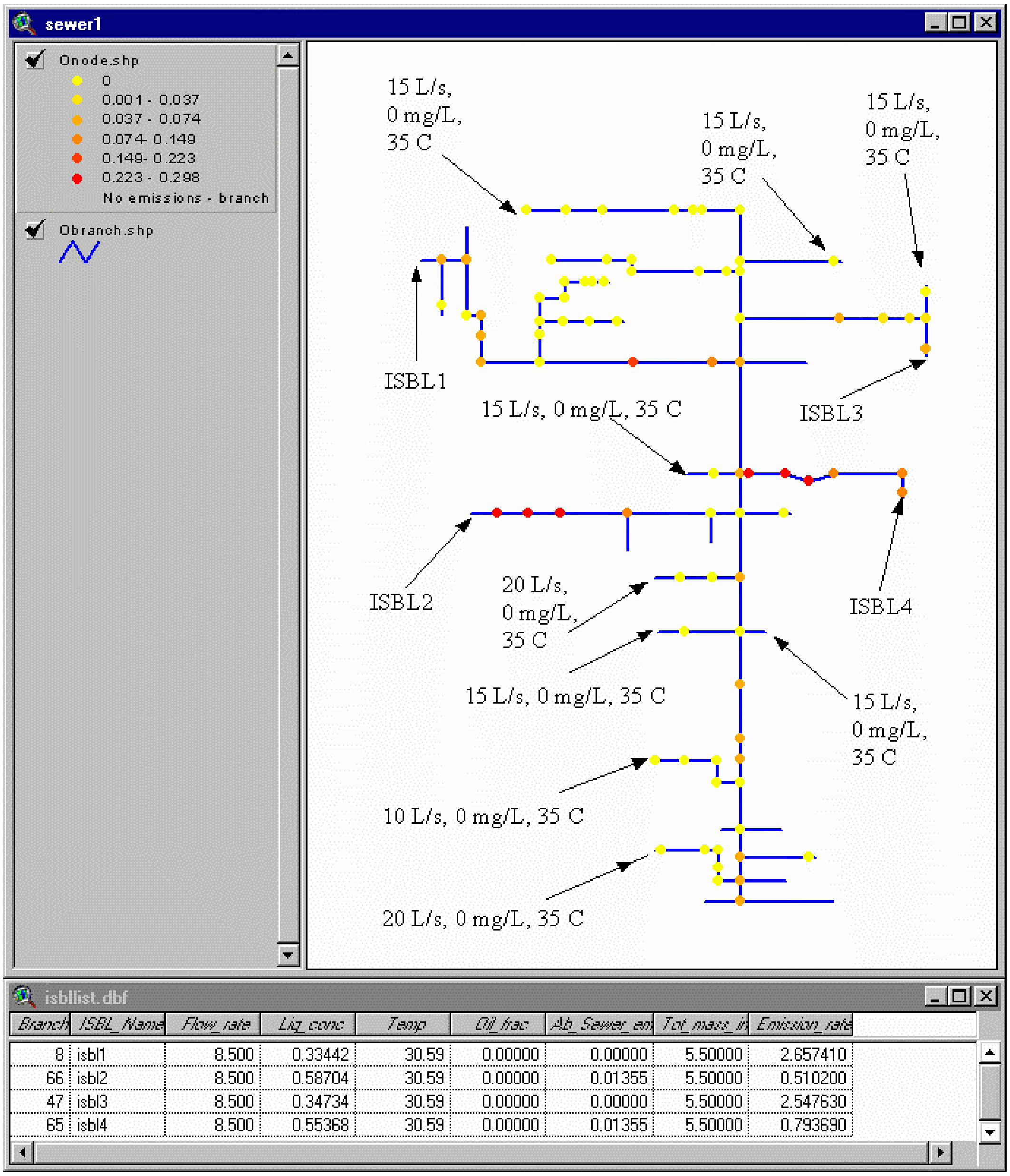
Implementation of the additional ranges allowed the distinction between nodes with low, non-zero emissions and nodes with zero emissions. Emissions were 0 mg/s for nodes where there was no mass entering from upstream points as well as at locations of junctions (no openings for air exchange). The new display ranges showed that emissions immediately downstream from ISBL1 and ISBL3 had low, non-zero values, as indicated by the light orange nodes.

Comparing the total nodal emissions with the total emissions at the ISBL units, fugitive emissions from throughout the OSBL system were seen to be significant. The total emission rate from the ISBL units was 6.51 mg/s and the total emission rate from nodes was 3.11 mg/s. The nodal emissions made up more than 32.3% of the total emissions (11.95 mg/s) from the combined system (ISBL and OSBL units). Table 4.5 shows the liquid concentration from each ISBL unit, as well as the emission rate from each ISBL. It also shows the total emissions attributed to the ISBL units (from drains and ISBL manholes and openings). Figure 4.14 shows the OSBL unit with all nodes with non-zero emissions labeled with the emission rate in mg/s. The nodes are colored by ranges corresponding to those shown in Figure 4.11.

The total mass of chemical discharged to the system was 22 mg/s. The total emission rate (9.62 mg/s) accounts for more than 40% of the chemical entering the sewer system. The contributions of both nodal emissions and ISBL emissions to the total rate from a sewer network indicated that both sources were significant.

Table 4.5: Liquid concentration discharged from each ISBL to the OSBL and emission rate from the ISBL.

ISBL	**Liquid Concentration** (mg/L)	**Emission Rate** (mg/s)
ISBL1	0.33	2.66
ISBL2	0.59	0.51
ISBL3	0.35	2.55
ISBL4	0.55	0.79
Total Emissions (mg/s)	6.51	



Branch	ISBL_Name	Flow_rate	Liq_conc	Temp	Oil_frac	Ab_Sewer_en	Tot_mass_in	Emission_rate
8	isbl1	8.500	0.33442	30.59	0.00000	0.00000	5.50000	2.657410
66	isbl2	8.500	0.58704	30.59	0.00000	0.01355	5.50000	0.510200
47	isbl3	8.500	0.34734	30.59	0.00000	0.00000	5.50000	2.547630
65	isbl4	8.500	0.55368	30.59	0.00000	0.01355	5.50000	0.793690

Figure 4.13: OSBL results shown with emphasis on low range emissions (emission rates from 0.0 mg/s to 0.074 mg/s). Nodes represented in yellow have no emissions. Red represents nodes with emissions in the highest range, with emissions rates increasing as the colors move from yellow through orange to red.

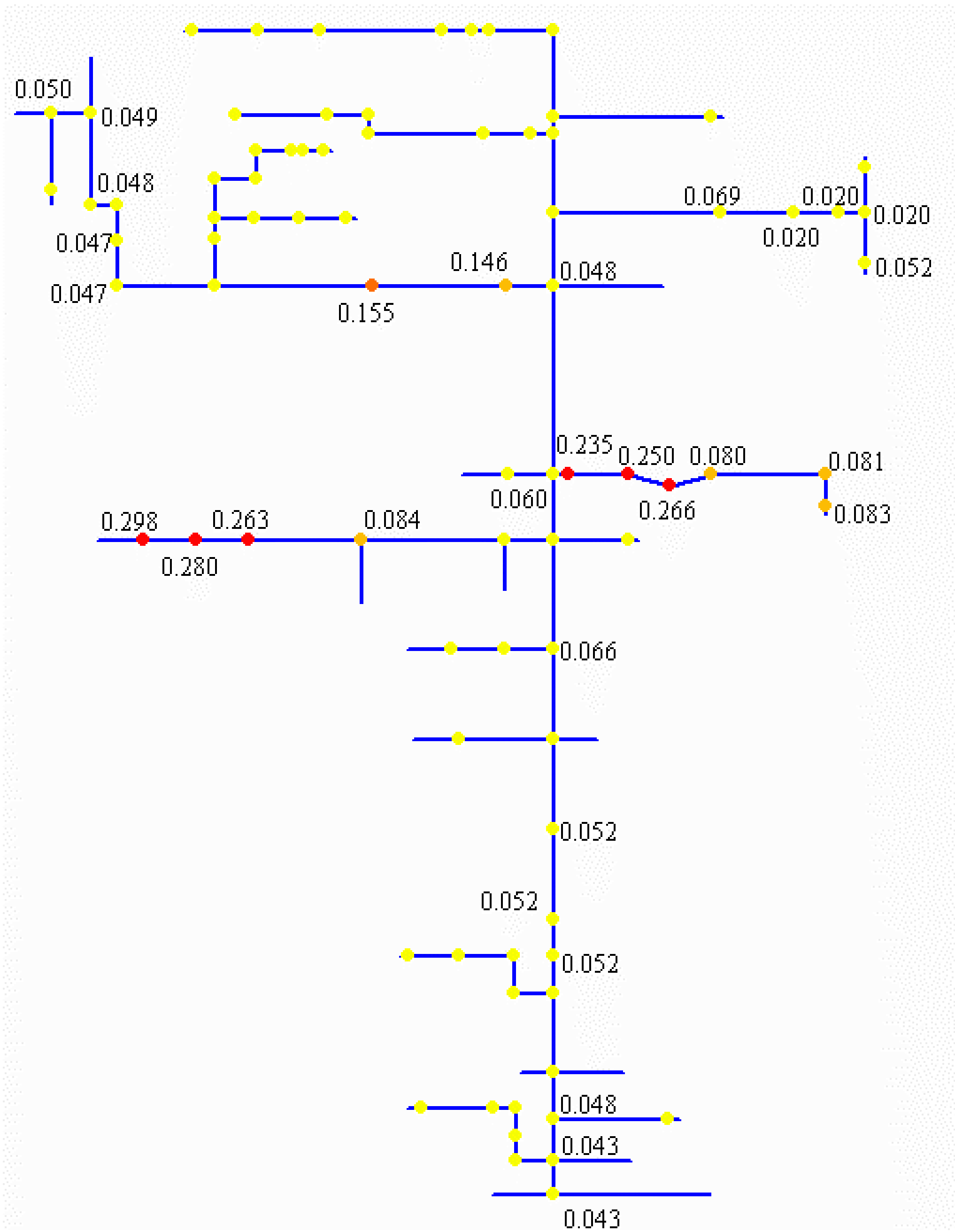


Figure 4.14: OSBL unit labeled by emission rates at nodes with non-zero emissions. The node colors correspond to those shown in Figure 4.11.

Chapter 5: Conclusions and Recommendations

5.1 SUMMARY

In this research, a procedure was created by which the naUTilus model could be run through the GIS software package, ArcView®. The procedure included digitizing schematic diagrams of the sewer network, storing the files in a specified format, processing the network in ArcView®, running naUTilus through ArcView®, and using ArcView® to analyze results. In addition to developing the procedure, this research included customizing the ArcView® environment through scripts to enable sewer processing, execution of naUTilus, and display of naUTilus results. The developed procedure and customized ArcView® environment were used to analyze an ISBL from a large refinery and an OSBL from a chemical manufacturing facility.

The developed procedure allowed both ISBL and OSBL analyses. It demonstrated naUTilus as a useful tool for estimating VOC emissions from industrial sewer networks. The enhancements gained through integrating naUTilus with ArcView® are listed below.

- A user-friendly interface is introduced. ArcView® prompts the user for all necessary data. This is one factor that facilitates the application of naUTilus to large sewer networks.The improved user interface made examination of varying conditions a more manageable task; effects of liquid temperature and ambient wind speed on VOC emissions were examined.

- ArcView® automates the creation of a naUTilus input file and automates execution of naUTilus. Automated creation of the input file lowers the chance that errors may be made during input file creation. Automated execution of naUTilus from the ArcView® environment allows all functions to be performed from one platform. Both factors facilitate application of naUTilus to large sewer networks
- Data storage is simplified. The specified data storage set up gives the user a framework for data storage, as well as enables the ArcView® scripts to locate all necessary files.
- ISBL units are connected to the OSBL unit. Output from naUTilus for the appropriate ISBL units is extracted for OSBL naUTilus input.
- Results are displayed visually. A visual display of OSBL results allowing easy "hot spot" identification. Display of other factors, in addition to emissions estimates, allow analysis factors influencing the level of emissions.
- A spatial reference is introduced. A spatial reference is instrumental not only in providing some of the other listed practical benefits (i.e. examining the effect of sealed drain placement), but also introduces potential benefits, such as the possiblity of incorporating fugitive emissions in fencline risk assessments.

Analysis of the ISBL unit was done for several conditions. These analyses yielded the following conclusions:

- Stripping efficiency increases with increasing liquid temperature. This is likely due to the effect of temperature on the Henry's law constant.
- Stripping efficiency increases with increasing ambient wind speed. This is due to higher ventilation rates. The rate of increase approaches zero for higher ambient wind speed.
- Stripping efficiency decreases with increasing liquid flow rate. This was consistent with the mass transfer theory incorporated in the naUTilus model. As liquid flow rate increases, hydraulic retention time decreases. This decreases the amount of time over which mass transfer can occur between the liquid and gas phases.
- Stripping efficiency decreases with an increasing number of sealed drains. This is caused by a decrease in ventilation.
- Stripping efficiency is affected by the location of sealed drains. This is seen by the scatter in the data when sealed drains are placed in groups throughout the ISBL unit. The effect of increased numbers of sealed drain had a more notable effect than sealed drain location.

Analysis of the OSBL unit done on a visual basis through the display of naUTilus results in ArcView®. Examining OSBL results led to the following conclusions:

- The algorithm used by naUTilus to calculate ventilation affects the location of emissions output by naUtilus.

- The location of emissions is also affected by changes in hydrodynamic conditions due to varying branch diameters.
- Nodal emissions throughout the OSBL unit may make up a significant fraction of the overall emissions from industrial sewers.
- Factors affecting emissions are more easily determined by applying ArcView® due to the capability to display emissions and other conditions visually throughout the sewer network.

5.2 RECOMMENDATIONS AND LIMITATIONS

The potential benefits of the spatial reference of naUTilus suggests that further work could be done to fully utilize it. In particular, a potential application involves the user of naUTilus results with dispersion models to perform fenceline risk assessments for hazardous air pollutants (HAPs). The spatial reference in ArcView® would allow identification of emission locations relative to the fenceline of the property.

Before applying naUTilus towards the goal of fenceline risk assessment, work should also be done to address other limitations of the project. Limitations of this research are due in part to assumptions made by the naUTilus model. For example, as discussed in Section 4.2, the algorithm used by naUTilus for ventilation heavily drives the location of emissions throughout the OSBL network. For this, and other reasons, validation of the naUTilus model is recommended.

Other possible uses of the naUTilus/ArcView® interface do not depend on a true geographic reference or accurate placement of emissions. It is still useful in estimating the quantity of emissions from an industrial sewer. By employing the improved user interface, analysis of emissions with respect to drops and inflow, similar to the analyses done on flow rate, temperature, and wind speed, could be used for planning sewer designs.

Appendix A

METHODS OF NUMBERING SEWER NETWORKS

MW Kellogg Method

MW Kellogg is a Houston-based company that plans and builds industrial facilities. Data describing sewer dimensions of the ISBL unit were numbered by the MW Kellogg method. MW Kellogg numbered each sewer element as part of one set, regardless of element type. The general numbering method is described in the following steps:

1. Create a simplified schematic.
2. Identify the main sewer line.
3. Identify each subsystem. Subsystems parts of the sewer network which feed to a manhole or junction box along the main sewer line.
4. Number elements starting from Subsystem A.

The MW Kellogg method of numbering sewer elements has three fundamental differences from the other two methods of numbering elements discussed in this thesis. First, all elements are numbered in one set, regardless of element type. Second, points at which two branches meet are not numbered unless they meet at a junction box. In both the naUTilus and GIS methods of numbering elements, any point where two branches meet is a numbered element. Third, cleanout connections are numbered elements represented in the system.

Figure A-1 illustrates part of the ISBL. Subsystem A and subsystem B are numbered by the MW Kellogg method. The ISBL unit consists of 11 subsystems and 395 numbered elements.

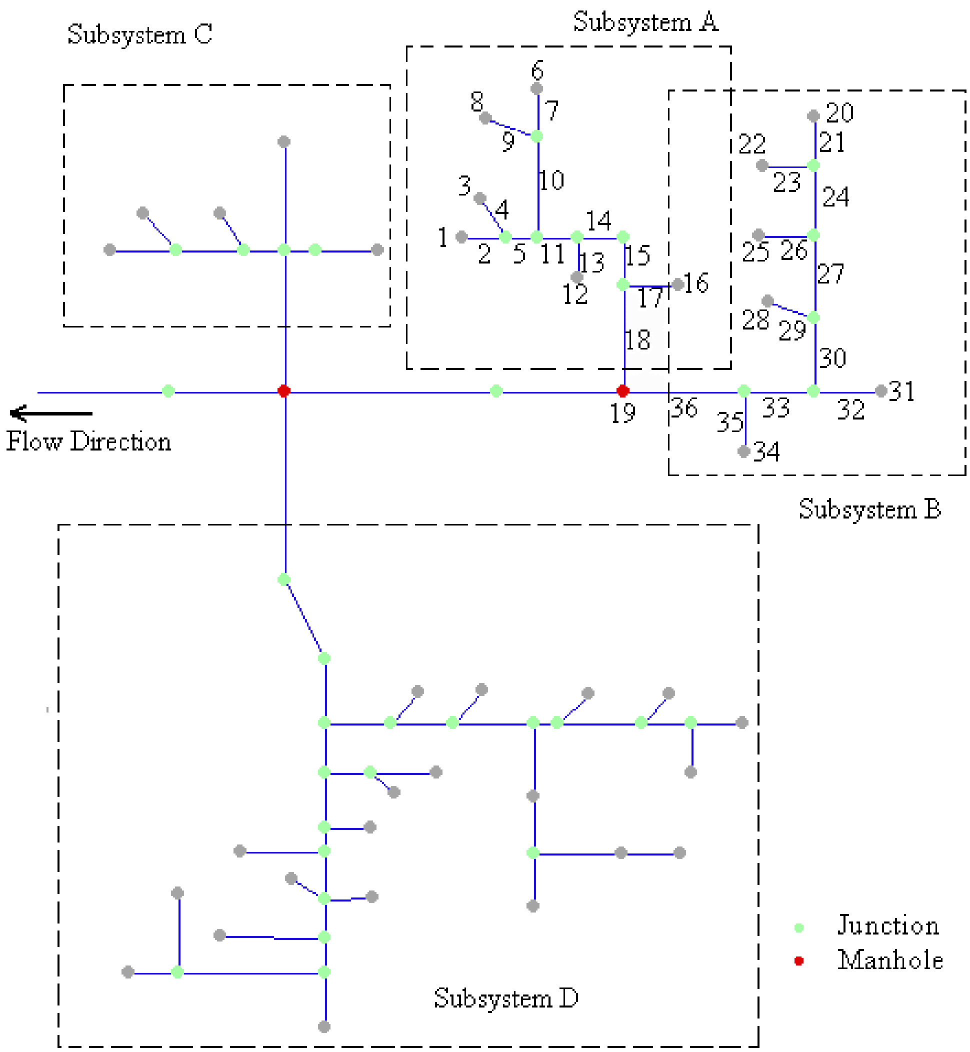


Figure A-1: Part of the ISBL unit numbered by the MW Kellogg numbering method.

naUTilus Method

When numbering sewer elements for naUTilus, two types of features are used to describe the sewer network: lines and points. Lines describe the branches of the sewer network. Points describe what naUTilus considers nodes. Elements described by nodes are manholes, junctions, and on-line drains. Other sewer elements are not numbered; their location is specified by their connection to a numbered element. For example, elbow drains in an ISBL occur at the initial end of a branch. The naUTilus numbering system does not number elbow drains, but associates them with the branches to which they connect. Thus, as seen in Figure A-2, the node labeled "Elbow drain A" is associated with branch 1.

For naUTilus to run, it is necessary to establish both element connectivity and flow connectivity. As discussed previously, the connectivity of an elbow drain is established by specifying the branch to which it is attached. Likewise, node to element connectivity is established for manholes, junctions, and on-line drains by specifying the node to which the element is connected. Establishing flow connectivity consists of specifying which branches flow into and out of each node. Accuracy in establishing flow connectivity is necessary to ensure mass balance closure and for naUTilus to run correctly. Flow connectivity is established node by node in the naUTilus input file. It is discussed for ISBL units in Section 3.3.3.1 and for OSBL units in Section 3.3.3.2.

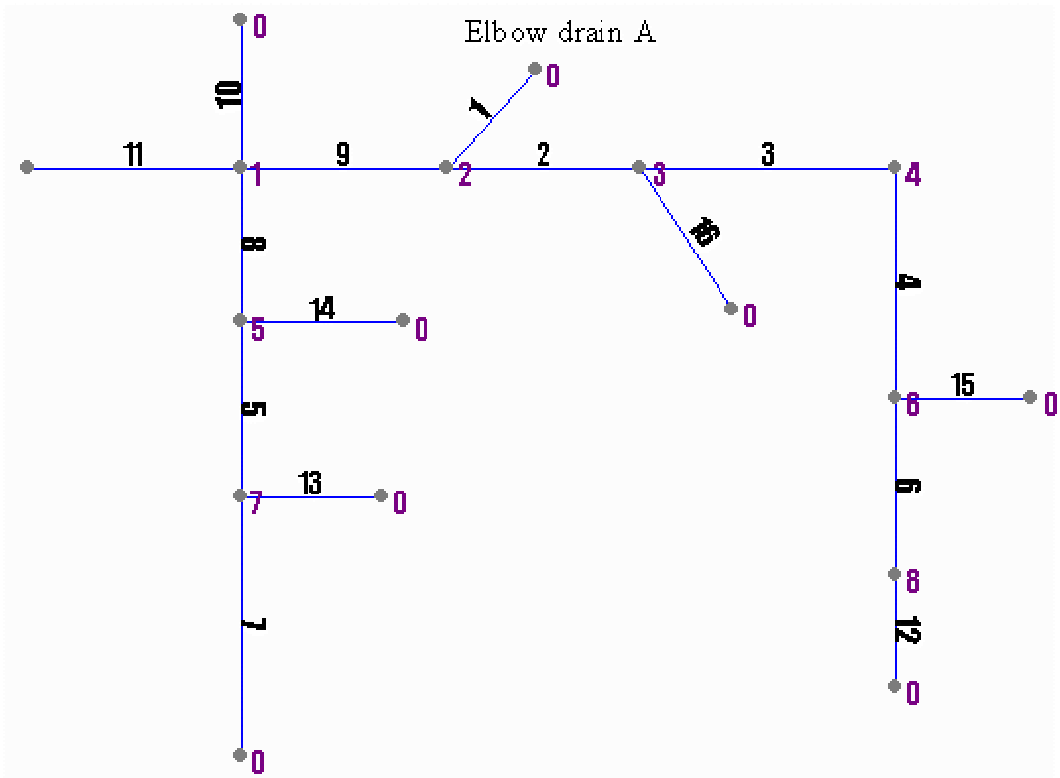


Figure A-2: Example of an ISBL unit numbered using the naUTilus numbering method. Nodes are numbered in purple, with elbow drains are indicated by "0". Branches are numbered in black.

GIS Method

In the GIS method of numbering elements, as in the naUTilus method, two sets of elements are used: lines and points. The lines represent the pipes in the sewer network and the points represent features naUTilus considers nodes, as well as elbow drains, which naUTilus does not number. The same ISBL unit numbered in Figure A-2 through the naUTilus method is numbered in the GIS method in Figure A-3.

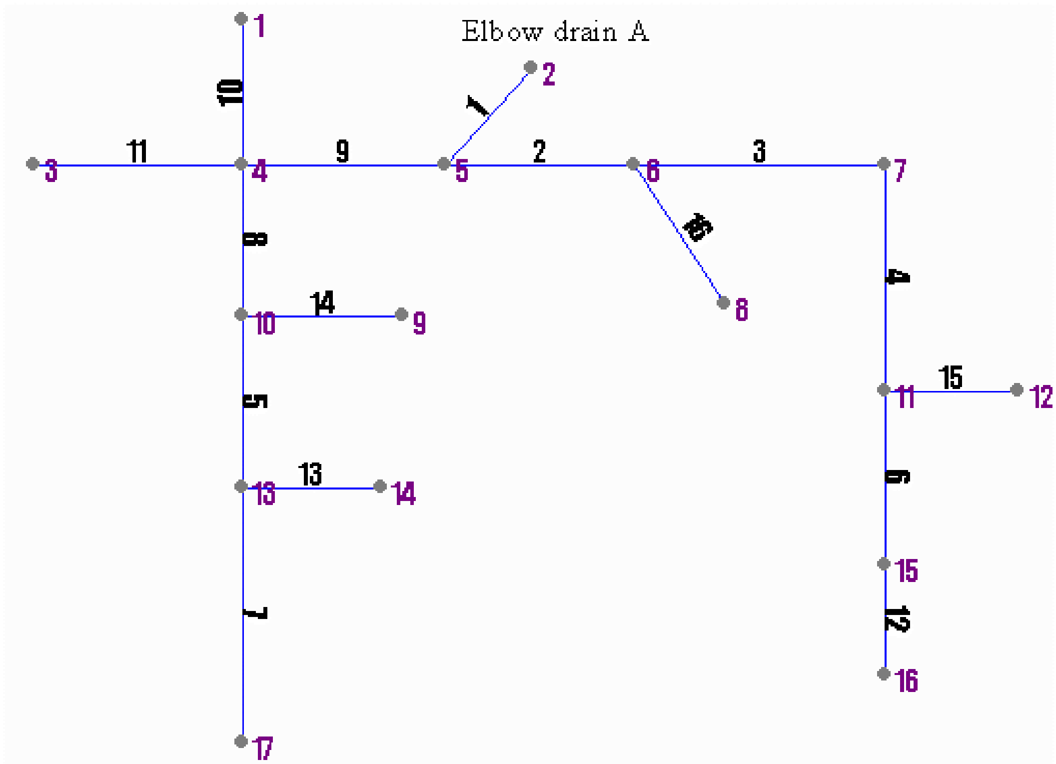


Figure A-3: Example of an ISBL unit numbered using the GIS method. Branches are numbered in black. Nodes and elbow drains are numbered in purple.

Appendix B

DIGITIZING AND PROCESSING SEWER NETWORKS

Comments on Digitizing a Sewer Network and Preprocessing Digitized Files

To apply GIS technology to industrial sewers, the sewer network must first be represented in a digital format, which can be done through digitizing. This can be done in AutoCad or an equivalent program. This document describes the method for digitizing a sewer network to ensure compatibility with the naUTilus/ArcView® connection. It also discusses the pre-processing of a digitized file, briefly discussed in Section 3.2, to establish connectivity.

Two methods of digitizing and processing are discussed. The first method involves an intermediate step in Arc/Info, a UNIX based GIS program produced by ESRI, to establish flow connectivity of the sewer elements. This method was used for this research. The second method, discussed briefly, is based solely on use of AutoCad and ArcView®.

Method 1: The Arc/Info Option

The first step towards integrating naUTilus and ArcView® is to represent a sewer network in a digital format by digitizing it in AutoCad or an equivalent program. Both the ISBL unit and the OSBL unit described in this research were digitized in AutoCad. As described in Section 2.1, industrial sewer networks consist of several elements, each of which can be represented by either a point or a line. In an ISBL network, drains, manholes, drops, and junctions are all point features. In an OSBL network, ISBL locations, manholes, drops, and junctions are point features. For both ISBL and OSBL units, the only line features are the sewer reaches.

Important considerations when digitizing an industrial sewer include the following:

- Digitize each ISBL unit and OSBL unit separately. This is done to create a unique file for each unit.
- Digitize lines and points in separate layers. This allows the layers to be opened in Arc/Info correctly.
- Digitize every point and line feature in the unit, including those not numbered in the naUTilus numbering method. This ensures a data set consistent with the provided schematic or drawing.
- Digitize each line in the direction of flow. This ensures flow direction in each branch is in the correct in the ArcView® project and ensures mass balance closure.
- Save each digitized file as a data exchange format (DXF) file.

Once the units of the sewer network have been digitized and saved as DXF files, the can be opened and processed in Arc/Info. The commands to process a DXF file called test.dxf are shown in Figure B-1. The commands are shown for an ISBL unit.   The command DXFARC creates a coverage from the DXF file. This coverage should be named IBRANCH. The command build is used to define the features in the coverage. ARCSHAPE is used to create two shape files from the coverage; one shape file describes the ISBL branches and another shape file describes the ISBL nodes.

```
Arc: dxfarc test.dxf ibranch

Enter layer names and options (type END or $REST when done)
============================================================
Enter the 1st layer and options : branches arc
Enter the 2nd layer and options :
Character string expected.
Done entering layer names and options (Y/N)? y
Do you wish to use the above layers and options (Y/N)? y

Processing /HOME1/ALPHA62/HOWCF/TEMP/TEST.DXF ...
No labels, killing XCODE...
Externalling BND and TIC...

         92 Arcs written.
          0 Labels written.
          0 Annotations written.
          0 Annotation levels.
Arc: build ibranch arc
 Building lines...
Arc: build ibranch node
 Building nodes...
Arc: arcshape ibranch arc ibranch
INFO item name FNODE# modified to dBASE field FNODE_
INFO item name TNODE# modified to dBASE field TNODE_
INFO item name LPOLY# modified to dBASE field LPOLY_
INFO item name RPOLY# modified to dBASE field RPOLY_
INFO item name IBRANCH# modified to dBASE field IBRANCH_
INFO item name IBRANCH-ID modified to dBASE field IBRANCH_ID
INFO table IBRANCH.aat copied to dBASE database ./ibranch.DBF
Fields: 7, Records: 92
Arc: arcshape ibranch node inodes
INFO item name ARC# modified to dBASE field ARC_
INFO item name IBRANCH# modified to dBASE field IBRANCH_
INFO item name IBRANCH-ID modified to dBASE field IBRANCH_ID
INFO table IBRANCH.nat copied to dBASE database ./inodes.DBF
Fields: 3, Records: 97
```

Figure B-1: Commands in Arc/Info used to process an ISBL unit. The resulting files are the shape files describing the ISBL.

The commands for an OSBL unit are similar to those for an ISBL unit. For OSBL units, the coverage extracted from the DXF file must be named OSBL. The command build is applied to this coverage for both nodes and arcs. Two shape files are created from the OSBL coverage using the command arcshape. A summary of the commands is given in Table B-1. Table B-2 lists the arguments for each command for ISBL and OSBL units.

Table B-1: Summary of Arc/Info commands used to process industrial sewers.

Command	Description
DXFARC	Creates an Arc/Info coverage from the DXF files. Form: Dxfarc <dxf_file_name.dxf> <coverage_name>
BUILD	Defines feature types in the coverage. Form: build <coverage_name> <feature_type> This should be done for arcs (lines) and nodes.
ARCSHAPE	Creates shape files out of an Arc/Info coverage. Form: arcshape <coverage_name> <feature_type> <shapefile_name>

Table B-2: Arguments for the commands listed in Table B-1 for ISBL and OSBL units. Note that two shape files are created for each unit, thus two values are listed. Two values are also listed for feature type. Each value is used once with the command "build" and once with "arcshape".

Argument	Unit Type	Value
<dxf_file_name>	ISBL or OSBL	Not specified
<coverage_name>	ISBL	IBRANCH
<coverage_name>	OSBL	OSBL
<feature_type>	ISBL or OSBL	ARC
<feature_type>	ISBL or OSBL	NODE
<shapefile_name>	ISBL	Ibranch, inodes
<shapefile_name>	OSBL	Obranch, onode

Alternative method

In the absence of access to Arc/Info, it may be necessary to digitize the sewer network in a different manner. Because the intermediate step in Arc/Info establishes the flow connectivity of the network, when this step is excluded connectivity must be assigned manually.

Connectivity can be assigned manually during the digitizing process. Though more demanding than the previously discussed method, it may be necessary in the absence of access to Arc/Info. If using this process, both nodes and branches must be digitized and manually numbered. Each branch must be attributed with a "from node" and a "to node." Each node must be attributed a branch consistent with the system through which Arc/Info assigns branches. ArcView® is equipped with a CAD reader, allowing the digitized map to be read in ArcView®. This process was used to represent the municipal sewers in East Baton Parish, Louisiana in a project done by Montgomery Watson's Baton Rouge office (Moeller, 1997).

Appendix C

AVENUE SCRIPTS

```
'
'------------------------------------------
'-----------Creation Information-----------
'------------------------------------------
'
'Name: Start
'Author:  Cindy How
'Date:    7/8/97
'
'------------------------------------------
'----------------Purpose-------------------
'------------------------------------------
'
'This script was created to explore and test
'the proposed file structure for the naUTilus/
'ArcView® system.  It should be run from the
'directory which holds the nautilus.APR file.  In
'this main directory, subdirectories should
'exist for each ISBL unit attached to the
'OSBL unit represented.  No other subdirectories
'should exist.
'
'This script finds all the subdirectories
'(assumed to hold an ISBL.apr project) and
'creates a table in the OSBL project which will
'or does hold the data on ISBL effluent.
'
'This table will be used to hold data for
'input to the OSBL unit from the ISBL units.
'
'This script will also create a view for the OSBL
'unit and each ISBL unit, titled by the name
'given to the directory in which the information
'is stored.  The OSBL view window will be opened.
'
'
'------------------------------------------
'---------Get Initial Information----------
'------------------------------------------


TheProject = av.GetProject
ProjName = TheProject.GetFileName.GetBaseName.AsString
PrName = ProjName.BasicTrim("",".apr")
'open OSBL view titled by project name
OView = View.Make
OView.SetName(PrName)
OSBLWin = Oview.GetWin
TheMainDir = FN.GetCWD

'Add OSBL Themes to OView
BrSrc = SrcName.Make("obranch.shp")   'create the sourceNames
NDSrc = SrcName.Make("onode.shp")
BrTheme = Theme.Make(BrSrc)           'make the themes
NdTheme = Theme.Make(NdSrc)
OView.AddTheme(BrTheme)               'Add the themes
OView.AddTheme(NdTheme)
BrTheme.SetVisible(true)              'display the themes
NdTheme.SetVisible(true)

'change appearance of themes
BrLegend = BrTheme.GetLegend
NdLegend = NdTheme.GetLegend
BrSymbs = BrLegend.GetSymbols
NdSymbs = Ndlegend.GetSymbols

Col1 = Color.GetBlue
Col2 = Color.GetGray
Brsymbs.UniformColor(Col1)
NdSymbs.UniformColor(Col2)

OSBLWin.Open


TheDir = FN.GetCWD
ISBLList = TheDir.Read("*.isb")

ListCount = ISBLList.Count
NameList = List.Make

For each i in 1..ListCount
  Idx = i - 1
  ISBLUnit = ISBLList.Get(idx)
  ISBLName = ISBLUnit.GetBaseName
  NameList.Add(ISBLName)
end
```

```
’create ODB for permanent variables
TheODB = ODB.Make("permdat.odb".AsFilename)
TheODB.Add(NameList)
TheODB.Add(TheMainDir)
TheODB.Commit

TabList = TheDir.Read("*.dbf")
TabCount = TabList.Count

For each i in 1..TabCount
  idx = i - 1
  Tabname = TabList.get(idx)
  Tname = Tabname.GetBaseName
  TabList.Set(idx,Tname)
end

FindTab = TabList.FindByValue("isbllist.dbf")

TabName = "isbllist.dbf".AsFilename
TheTab = Vtab.MakeNew(TabName,dBase)
BrFld = Field.Make("Branch",#Field_short,4,0)
NFld = Field.Make("ISBL_Name",#Field_Char,10,0)
ASFld = Field.Make("Flow_rate",#Field_decimal,10,3)
SEFld = Field.Make("Liq_conc",#Field_decimal,10,5)
MIFld = Field.Make("Temp",#Field_decimal,10,2)
ERFld = Field.Make("Oil_frac",#Field_decimal,10,5)
TheTab.AddFields({BrFld,NFld,ASFld,SEFld,MIFld,ERFld})
for each i in 1..ListCount
  TheTab.AddRecord
  Idx = i - 1
  IName = NameList.Get(Idx).AsString
  TheTab.SetValue(NFld,Idx,IName)
end


TheTable = Table.Make(TheTab)
TheTable.SetName("isbllist.dbf")

TheTab.SetEditable(False)

’------------------------------------------
’-----------Add ISBL Views-----------------
’------------------------------------------
OldDir = FN.GetCWD


For each record in TheTab
  NewView = View.Make
  ViewName = TheTab.ReturnValue(NFld,record)
  NewView.SetName(ViewName)
  NewWin = NewView.GetWin
  TheDir = ViewName.AsFilename
  TheDir.SetCWD
  IBrSrc = SrcName.Make("ibranch.shp")   ’create the sourceNames
  INDSrc = SrcName.Make("inodes.shp")
  IBrTheme = Theme.Make(IBrSrc)          ’make the themes
  INdTheme = Theme.Make(INdSrc)
  NewView.AddTheme(IBrTheme)             ’Add the themes
  NewView.AddTheme(INdTheme)
  IBrTheme.SetVisible(true)              ’display the themes
  INdTheme.SetVisible(true)

  TheODB = ODB.Make("isbl.odb".AsFilename)  ’create ODB file
  TheODB.Commit

  av.Run("inittabs",{ViewName})

  OldDir.SetCWD    ’Return to main directory

   ’change appearance of themes
  IBrLegend = IBrTheme.GetLegend
  INdLegend = INdTheme.GetLegend
  IBrSymbs = IBrLegend.GetSymbols
  INdSymbs = INdlegend.GetSymbols

  Col1 = Color.GetBlue
  Col2 = Color.GetGray
  IBrsymbs.UniformColor(Col1)
  INdSymbs.UniformColor(Col2)


end

’------------------------------
’--------END SCRIPT------------
’------------------------------
```

```
'-------------------------------------------
'------------Creation Information-----------
'-------------------------------------------
'Name: Inittabs
'Created by Cindy How
'
'Purpose:
'To initiate tables for ISBL units
'The tables created are empty tables
'to be filled through other scripts.
'This script is called by the script
'START.

TheProject = av.GetProject
TheViewName = Self.Get(0)
TheView = TheProject.FindDoc(TheViewName)
TheWin = TheView.GetWin
TheName = TheWin.GetName.AsFilename
TheDir = TheName.SetCWD

av.Run("sortpt",{TheViewName})
av.Run("tables",{TheViewName})
'av.Run("connctit",{TheViewName}) 'need to change (run later in process!)
av.Run("drtab",{TheViewName})

DTab = av.FindDoc("drains.dbf")
'CTab = av.FindDoc("nodecon.dbf")
BrTab = av.Finddoc("brtable.dbf")
Drop = av.FindDoc("drops.dbf")
HPTab = av.FindDoc("hardpipe.dbf")

TheProject.RemoveDoc(DTab)
'TheProject.RemoveDoc(CTab)
TheProject.RemoveDoc(BrTab)
TheProject.RemoveDoc(Drop)
TheProject.RemoveDoc(HPTab)

'------------------------------
'--------END SCRIPT------------
'------------------------------
```



```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: sortpoint
'Author:  Cindy How
'Date:    4/97
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take all points in the inodes.shp theme
'and find which are terminal points (elbows vs online drains
'or manholes).
'
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  Attribute table for the theme nodes
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
'1.  Added field to node theme, indicating elbow (-1)
'    or online drain or manhole (numbered)
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
ViewName = Self.Get(0)
TheView = TheProject.FindDoc(ViewName)
TheTheme= TheView.FindTheme("inodes.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.FindField("Inodes")
TheFtab.SetEditable(True)
BrTheme = TheView.FindTheme("ibranch.shp")
BrTab = BrTheme.GetFtab
TNodeField=BrTab.FindField("Tnode_")
FnodeField = BrTab.FindField("Fnode_")
```

```
’-----------------------------------------------
’-----Add New Fields to Node Attribute Table----
’-----------------------------------------------
’check for prior existance of fields
TypeFld = TheFtab.FindField("Node_type")
NodeFld = TheFtab.FindField("Node")
NTypeFld = TheFtab.FindField("Manhole")
DFld = TheFtab.FindField("Drain")
TFld = TheFtab.FindField("Type")
FldList = List.Make
If (TypeFld.AsString = "nil") then
  TypeFld=Field.make("Node_Type",#Field_short,4,0)
  FldList.Add(TypeFld)
else
end


If (NodeFld.AsString = "nil") then
  NodeFld=Field.make("Node",#Field_short,4,0)
  FldList.Add(NodeFld)
else
end


If (NTypeFld.AsString = "nil") then
  NTypeFld = Field.make("Manhole",#Field_short,4,0)
  FldList.Add(NTypeFld)
else
  For each record in TheFtab
    TheFtab.SetValue(NtypeFld,record,0)
  end
end


If (DFld.AsString = "nil") then
  DFld = Field.Make("Drain",#Field_short,4,0)
  FldList.Add(DFld)
else
  For each record in TheFtab
    TheFtab.SetValue(DFld,record,0)
  end
end
If (TFld.AsString = "nil") then
  TFld = Field.Make("Type",#Field_char,16,0)
  FldList.Add(TFld)
else
end


TheFtab.setEditable(true)
TheFtab.AddFields(FldList)


TNodeList = List.Make ’Makes List of "to Nodes"
Fnodelist = list.make ’makes a list of "from nodes"
i=0
For each rec in BrTab
  TNode = BrTab.ReturnValue(TNodeField,i)
  TNodeList.Add(Tnode)
  Fnode = BrTab.ReturnValue(FnodeField,i)
  Fnodelist.Add(Fnode)
    i = i +1
end


j= 1  ’Assigns value of -1 to the field "Node_Type" for terminal nodes
For each rec in TheFtab
  Check = TNodeList.FindbyValue(j)
      If (check = -1) then
      TheFtab.SetValue(TypeFld,rec,check)
    else
      TheFtab.SetValue(TypeFld,rec,1)
    end
  Check2 = FnodeList.FindByValue(j)
    If (check2 = -1) then
      TheFtab.SetValue(typeFld,rec,2) ’set type value of last node in system to 2
(last node by ArcView® system)
    Else
    end
    j = j+1
end


k = 1
for each rec in TheFtab
  type = TheFtab.ReturnValue(TypeFld,rec)
```

```
  if (type = -1) then
  elseif (type = 2) then
    TheFtab.SetValue(NodeFld,rec,-1)
  else
    TheFtab.SetValue(NodeFld,rec,k)
    k=k+1
  end
  TheFtab.SetValue(TFld,rec,"Drain")
end

TheFtab.SetEditable(false)
'-----------------------------
'--------END SCRIPT------------
'-----------------------------
```



```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: tables
'Author:  Cindy How
'Date:    7/16/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script was created to combine parts of
'several scripts which created tables to be used
'in creating the naUTilus input file.  To reduce
'the number of buttons and scripts run separately,
'this script will create the tables and perform
'as many functions as possible towards filling
'those tables.  Not all functions will be possible
'as the order in which some things are run inhibits
'the complete consolidation of all scripts.
'
'This must be run from the View Window via button
'for each ISBL
'
```

```
'---------------------------------------------
'------------------Input----------------------
'---------------------------------------------
'
'1.  Node theme (inodes.shp)
'2.  Attribute table of node theme
'3.  Branch theme (ibranch.shp)
'4.  Attribute table of branch theme
'
'---------------------------------------------
'-----------------Output----------------------
'---------------------------------------------
'1.  New table called hardpipe
'2.  New table called drops
'3.  New table called drains
'4.  New table called brtable
'5.  New table called nodecon.dbf
'
'Note that some of these tables will be empty
'after running this script.
'
'---------------------------------------------
'---------Getting initial data-----------------
'---------------------------------------------
theProject=av.getProject
TheViewName = Self.Get(0)
TheView = TheProject.FindDoc(TheViewName)
TheWin = TheView.GetWin
TheName = TheWin.GetName.AsFilename
TheDir = TheName.SetCWD
NodeTheme = TheView.FindTheme("inodes.shp")
NodeFtab = NodeTheme.GetFtab
BranchTheme = TheView.FindTheme("ibranch.shp")
BrFtab = BranchTheme.GetFtab

'---------------------------------------------
'---------------Create Tables-----------------
'---------------------------------------------

'Hard Pipe Table
TabName="hardpipe.dbf".AsFilename
HPTable=VTab.MakeNew(TabName,dBase)

CFld=Field.make("Connect",#Field_long,8,0)
```

```
FFld=Field.make("Flow_rate",#Field_decimal,8,4)
TFld=Field.make("Temp",#Field_decimal,8,2)
ConFld = Field.Make("Conc",#Field_decimal,8,4)
OilFld = Field.Make("Oil_fraction",#Field_decimal,8,4)
HPTable.AddFields({CFld,FFld,TFld,ConFld,OilFld})

TheTab = Table.Make(HPTable)
TheTab.SetName("hardpipe.dbf")

TheODB = ODB.Open("isbl.odb".AsFilename)
TheODB.Add(TheTab)
TheODB.Commit

'Drop Table
Tab2Name="drops.dbf".AsFilename
DropTable=VTab.MakeNew(Tab2Name,dBase)

NFld=Field.make("Node",#Field_long,8,0)
BFld=Field.make("Branch",#Field_long,8,0)
HFld=Field.make("Height",#Field_decimal,10,4)
TFld=Field.make("Tailwater",#Field_decimal,10,4)
DropTable.AddFields({NFld,BFld,HFld,TFld})

TheTab2 = Table.Make(DropTable)
TheTab2.SetName("drops.dbf")

TheODB = ODB.Open("isbl.odb".AsFilename)
TheODB.Add(TheTab2)
TheODB.Commit

'Branch Table
LFld = BrFTab.FindField("Length")
NewName = "brtable.dbf".AsFilename
BrTab = VTab.MakeNew(NewName,dBase)

NewLFld = Field.Make("Length",#Field_Float,8,4)
DiamFld = Field.Make("Diameter",#Field_Float,8,4)
SlopeFld = Field.Make("Slope",#Field_Float,8,4)
BrTab.AddFields({NewLfld,DiamFld,SlopeFld})
BrTab.SetEditable(true)

DDiam = 0.15337
Dslope = 0.01
```



```
For each record in BrFTab 'fill table with default values
  BrTab.AddRecord
  brlength = BrFTab.ReturnValue(LFld,record)
  BrTab.SetValue(NewLfld,record,brlength)
  BrTab.SetValue(DiamFld,record,DDiam)
  BrTab.SetValue(SlopeFld,record,DSlope)
end

BrTab.SetEditable(False)
TheTab = Table.Make(BrTab)
TheTab.SetName("brtable.dbf")

TheODB = ODB.Open("isbl.odb".AsFilename)
TheODB.Add(TheTab)
TheODB.Commit

'---------------------------------------------
'---------------------END----------------------
'---------------------------------------------
```

```
’----------------------------------------------
’---------Creation Information-----------------
’----------------------------------------------
’Name: drtab
’Author:  Cindy How
’Date:    5/97
’Revisions:
’1.  7/15/97 by Cindy How
’    -Added lines to add the created table to
’     the ODB for the ISBL unit.
’----------------------------------------------
’------------Purpose/Description---------------
’----------------------------------------------
’This script create the drain input table for ISBL units
’
’Note: Elbow drains run to branches and on line
’drains are connected to nodes.
’
’-----------------------------------------------
’-------------------Input-----------------------
’-----------------------------------------------
’
’1.  Attribute table for the theme inodes
’2.  Attribute table for the theme ibranch
’
’-----------------------------------------------
’------------------Output-----------------------
’-----------------------------------------------
’
’1.  New table called drains.dbf to hold the information
’    on drains needed to create part of the naUTilus
’    input file.  This table is empty.
’-----------------------------------------------
’---------Get Initial Information---------------
’-----------------------------------------------
TheProject = av.GetProject
ViewName = Self.Get(0)
TheView = TheProject.FindDoc(ViewName)
TheTheme = theView.FindTheme("inodes.shp")
NodeTab = theTheme.GetFtab
BrTheme = theView.Findtheme("ibranch.shp")
BrTab = BrTheme.GetFtab
TypeFld = NodeTab.FindField("Node_Type")
NodeFld = NodeTab.FindField("Node")
BranchFld = NodeTab.FindField("Arc_")
InodeFld = NodeTab.FindField("Ibranch_")
InodeFld.setAlias("Inode")

TNodeFld = BrTab.FindField("Tnode_")
FnodeFld = BrTab.FindField("Fnode_")
’----------------------------------------------
’-----Define fields for and create new table----
’----------------------------------------------
TabName = "drains.dbf".asFilename
InTab = Vtab.MakeNew(TabName,dBase)
RefFld = Field.make("ISBL_ref",#Field_short,4,0) ’inodes.shp "inodes" number
for reference
DtypeFld = Field.make("Drain_Type",#Field_short,4,0)
DConnectFld = Field.make("Connectivity",#Field_short,4,0)
DFlowFld = Field.Make("Flow_rate",#Field_double,8,4)
DtempFld = Field.make("Temperature",#Field_float,8,2)
DconcFld = Field.Make("Concentration",#Field_double,8,4)
DdiamFld = Field.Make("Diameter",#Field_double,8,4)
DsealFld = Field.Make("Sealed",#Field_short,4,0)
DOilFld = Field.Make("Oil_frac",#Field_double,8,4)
InTab.AddFields({RefFld,Dtypefld,dconnectfld,dflowfld,dtempfld,dconcfld,ddia
mfld,dsealFld,doilFld})

TheTab = Table.Make(InTab)
TheTab.SetName("drains.dbf")

’Add table to the ODB file
TheODB = ODB.Open("isbl.odb".AsFilename)
TheODB.Add(TheTab)
TheODB.Commit
’-----------------------------------------------
’---------------------END-----------------------
’-----------------------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: iprompt0
'Author:  Cindy How
'Date:    7/23/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take use the node and branch
'theme of the ISBL system and help to create the input
'file by describing the connectivity as naUTilus
'requires.
'
'This script allows the user to specify if the
'majority of nodes are junctions or online drains
'and allows the user to specify those which
'are not among the majority.
'
'This script should be run from the view window
'of the ISBL unit which is being worked with.
'It should be run in conjunction with drjunc.
'
'----------------------------------------------
'------------------Input----------------------
'----------------------------------------------
'
'1.  Attribute table for the theme inodes
'2.  Attribute table for the theme ibranch
'
'----------------------------------------------
'------------------Output----------------------
'----------------------------------------------
'
'1.  Filled column in the inodes attribute table
'   indicating drain or junction (1 = drain,
'   -1 = junction)
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
NodeTheme = TheView.FindTheme("inodes.shp")
NodeTab = NodeTheme.GetFtab
TheField = NodeTab.FindField("Drain")
TheField2 = NodeTab.FindField("Node_type")
NodeTab.SetEditable(True)

ResetFlag = MsgBox.LongYesNo("Reset existing drain/junction data?","User
Select",true)
If (ResetFlag.AsString = "true") then
  ChoiceList = {"Drains (Unsealed)","Junctions","Drain/Junction (Internal nodes
default to junction)"}
  NdDef = MsgBox.ChoiceAsString(ChoiceList,"Select the default node
setting.","User Input")
  NdIdx = ChoiceList.FindByValue(NdDef)

  If (NdIdx = 0) then
    For each record in NodeTab
      NodeTab.SetValue(TheField,record,1)
    end
  elseif (NdIdx = 1) then
    For each record in NodeTab
      NodeTab.SetValue(TheField,record,-1)
    end
  elseif (NdIdx = 2) then
    For each record in NodeTab
      Flag = NodeTab.ReturnValue(TheField2,record)
      if (Flag = -1) then
        NodeTab.SetValue(TheField,record,1)
      else
        NodeTab.SetValue(TheField,record,-1)
      end
    end
  else
    msgbox.info("No choice made.  Rerun script.","Error")
    exit
  end

  TField = NodeTab.FindField("Type")


  TheLegend = NodeTheme.GetLegend
```

```
TheLegend.Load("sewerleg.avl".AsFilename,#Legend_loadtype_all)
For each record in NodeTab
  TheType = NodeTab.ReturnValue(TheField,record)
  if (TheType = 1) then
    NodeTab.SetValue(TField,record,"Drain (Unsealed)")
  else
    NodeTab.SetValue(TField,record,"Junction")
  end
end

NodeTab.SetEditable(False)
NodeTheme.UpdateLegend

if (nddef = "Drains (Unsealed)") then
  ndstr = "Junctions"
else
  ndstr = "Drains"
end

’MsgBox.Info("Please click on the nodes which are "++ndstr,"User Input")

else
end

MsgBox.Info("Please click on the nodes you wish to edit","User Input")

’------------------------------------
’----------END SCRIPT----------------
’------------------------------------
```



```
’---------------------------------------------
’---------Creation Information-----------------
’---------------------------------------------
’
’Name: drjunc
’Author:  Cindy How
’Date:    7/23/97
’
’---------------------------------------------
’------------Purpose/Description---------------
’---------------------------------------------
’
’This script will take use the node and branch
’theme of the ISBL system and help to create the input
’file by describing the connectivity as naUTilus
’requires.
’
’This script allows the user to specify if the
’majority of nodes are junctions or online drains
’and allows the user to specify those which
’are not among the majority.
’
’This script should be run from the view window
’of the ISBL unit which is being worked with.
’It should be run in conjunction with iprompt0.
’
’----------------------------------------------
’-------------------Input-----------------------
’----------------------------------------------
’
’1.  Attribute table for the theme inodes
’2.  Attribute table for the theme ibranch
’
’----------------------------------------------
’------------------Output-----------------------
’----------------------------------------------
’
’1.  Filled column in the inodes attribute table
’    indicating drain or junction (1 = drain,
’    -1 = junction)
’
’----------------------------------------------
’---------Get Initial Information---------------
’----------------------------------------------
```

```
TheProject = av.GetProject
TheView = av.GetActiveDoc
NodeTheme = TheView.FindTheme("inodes.shp")
TheFtab = NodeTheme.GetFTab
TheField = TheFtab.FindField("Drain")
TypeFld = TheFtab.FindField("Type")
MHField = TheFtab.FindField("Manhole")
TheFtab.SetEditable(true)


NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Nodetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
    Msgbox.error("That point is not found, try again","Not found")
    exit
  end
rec=PtOnTheme.get(0)

NdType = TheFtab.ReturnValue(TheField,rec)
'1 = drain, -1 = junction


MHFlag = TheFtab.ReturnValue(TypeFld,rec)
If (MHFlag = "Manhole") then
  TheList = {"Drain (Unsealed)","Junction","Manhole (no change)"}
  TheChoice = MsgBox.ChoiceAsString(TheList,"Reset from Manhole to:","User
Input")
  TheIndex = TheList.FindByValue(TheChoice)

If (TheIndex = 0) then
    TheFtab.SetValue(TheField,rec,1)  'drain
    TheType = "Drain (Unsealed)"
    TheFtab.SetValue(MHField,rec,0)
  elseif (TheIndex = 1) then
    TheFtab.SetValue(TheField,rec,-1)  'junction
    TheType = "Junction"
    TheFtab.SetValue(MHField,rec,0)
  else
    TheFtab.SetValue(TheField,rec,-1)  'manhole (cancel change)
    TheType = "Manhole"
  end

else
  NdType = NdType * -1
  TheFtab.SetValue(TheField,rec,NdType)
  If (NdType = 1) then
    TheType = "Drain (Unsealed)"
  else
    TheType = "Junction"
  end
end

TheLegend = NodeTheme.GetLegend

NodeTheme.UpdateLegend

TheFtab.SetValue(TypeFld,rec,TheType)

TheFtab.SetEditable(False)

'--------------------------------
'----------END SCRIPT------------
'--------------------------------




'---------------------------------
'-------Creation Information-------
'---------------------------------
'
'Name: Iprompt1
'Created by:  Cindy How
'Purpose: To prompt the user to select a node
'  which represents a manhole.
'To be run in conjunction with the script
'SELECT.

msgbox.info("Click on each node which represents a manhole.","User Input")

'---------------------------------
'-------------END SCRIPT-----------
'---------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: select
'Author:  Cindy How
'Date:    5/97
'Revisions:
'  1.  7/29/97 by Cindy How
'      - to be run with iprompt1
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take select the point to be
'flagged as a manhole.
'----------------------------------------------
'------------------Input-----------------------
'----------------------------------------------
'
'1.  Node theme
'2.  Attribute table of the Node theme
'
'----------------------------------------------
'-----------------Output-----------------------
'----------------------------------------------
'
'1.  Selected record
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme= TheView.FindTheme("inodes.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.Findfield("Manhole")
TypeFld =  TheFtab.FindField("Type")
DrFld = TheFtab.FindField("Drain")
ShpFld = TheFtab.FindField("Shape")
TheFtab.SetEditable(true)
NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Thetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
    Msgbox.error("That point is not found, try again","Not found")
    exit
  end
rec=PtOnTheme.get(0)

ManholeFlag = TheFtab.ReturnValue(TheField,rec)
If (ManholeFlag = 1) then
  ChoiceList = {"Drain","Junction"}
  ChoiceFlag = MsgBox.ChoiceAsString(ChoiceList,"Reset node: select the node
type","User Input")
  ChoiceIdx = ChoiceList.FindByValue(ChoiceFlag)
  If (ChoiceIdx = 0) then
    TheFtab.SetValue(TypeFld,rec,"Drain (Unsealed)")
    TheFtab.SetValue(DrFld,rec,1)
    TheFtab.SetValue(TheField,rec,0)
  elseif (ChoiceIdx = 1) then
    TheFtab.SetValue(TypeFld,rec,"Junction")
    TheFtab.SetValue(DrFld,rec,-1)
    TheFtab.SetValue(TheField,rec,0)
  else
    MsgBox.Info("No change made.","User cancel")
  end
else
  TheFtab.SetValue(TypeFld,rec,"Manhole")
  TheFtab.SetValue(DrFld,rec,-1)
  TheFtab.SetValue(TheField,rec,1)
end

TheTheme.UpdateLegend

TheFtab.SetEditable(false)

'----------------------------------------------
'---------------------END SCRIPT---------------
'----------------------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: iprompt2
'Author:  Cindy How
'Date:  7/23/97
'Editted: 11/20/97
' 1.  Allowed changing of drain sealed/unsealed status
'      without losing flow data, manholes/junctions
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This Script is to be run as the Click event with
'the Apply event DRAINDATA.  Most of the script
'is similar or identical to the previously written
'DRINPUT, altered to differentiate between drains
'and junctions.
'
'This script will distinguish elbow drains from
'on line drains by their location in the system.
'Elbow drains run to branches and on line drains
'are connected to nodes when connectivity is
'established for naUTilus.
'
'---------------------------------------------
'-------------------Input----------------------
'---------------------------------------------
'
'1.  Attribute table for the theme inodes.shp
'2.  Attribute table for the theme ibranch.shp
'3.  drains.dbf table (found in correct ISBL dir)
'    -empty (no records)
'---------------------------------------------
'------------------Output----------------------
'---------------------------------------------
'
'1.  drains.dbf table with one record per drain
'
'---------------------------------------------
'----------Get Initial Information-------------
'---------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc

'node theme
TheTheme = TheView.FindTheme("inodes.shp")
TheFtab = TheTheme.GetFtab
TypeFld = TheFtab.FindField("Node_Type")
NodeFld = TheFtab.FindField("Node")
BranchFld = TheFtab.FindField("Arc_")
InodeFld = TheFtab.FindField("Ibranch_")
InodeFld.setAlias("Inode")
TheField = TheFtab.FindField("Drain")
TFld = TheFtab.FindField("Type")
TheFtab.SetEditable(true)

'branch theme
BrTheme = TheView.FindTheme("ibranch.shp")
BrTab = BrTheme.GetFtab
TNodeFld = BrTab.Findfield("Tnode_")
FNodeFld = BrTab.FindField("Fnode_")


TheMainDir = FN.GetCWD
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
'msgbox.info("The CWD is"++TheName.AsString,"test")


TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(3)
TheProject.AddDoc(TheDoc)
Tabname = av.GetProject.FindDoc("drains.dbf")
InTab = TabName.GetVtab
InTab.SetEditable(True)
'find fields in Drains.dbf
RefFld = InTab.FindField("ISBL_ref")
DTypeFld = InTab.FindField("Drain_Type")
DConnectFld = InTab.FindField("Connectivity")
DFlowFld = InTab.FindField("Flow_rate")
DtempFld = InTab.FindField("Temperature")
DconcFld = InTab.FindField("Concentration")
DdiamFld = InTab.FindField("Diameter")
```

```
DsealFld = InTab.FindField("Sealed")
DOilfld = InTab.FindField("Oil_frac")

'---------------------------------------------
'---Fill drain table with connectivity data----
'---------------------------------------------

'check for already existing records
count = InTab.GetNumRecords
if (count > 0) then
  ChList = {"Reset drains","Edit existing drain data"}
  DrOpt = MsgBox.ChoiceAsString(ChList,"Would you like to reset all drains or
edit the existing drain data?","User Input")
  ChIndex = ChList.FindByValue(DrOpt)

else
  ChIndex = 1
end

if ((count > 0) and (chIndex = 0)) then
  For each record in InTab
      InTab.RemoveRecord(record)
  end
else
end

If ((count = 0) or (chIndex = 0)) then

  For each record in TheFtab
    DrFlag = TheFtab.ReturnValue(TheField,record)
    TermNodeFlag = TheFTab.ReturnValue(TypeFld,record)
    if (DrFlag = 1) then
      if (TermNodeFlag < 2) then
        recNum = InTab.AddRecord
        Inode = TheFtab.ReturnValue(InodeFld,record)
        InTab.SetValue(RefFld,recnum,Inode)
        TypeCheck = TheFtab.ReturnValue(TypeFld,record)
        If (TypeCheck = -1) then 'Elbow Drain
          InTab.SetValue(DTypeFld,recnum,1)
          ToBranch = TheFtab.ReturnValue(BranchFld,record)
          InTab.SetValue(DconnectFld,recnum,ToBranch)
        else
          InTab.SetValue(DtypeFld,recnum,2) 'On line drain
          Connode = TheFtab.ReturnValue(NodeFld,record)
          InTab.SetValue(DconnectFld,recnum,connode)
        end
      else
      end
    else
    end
  end
else
end


'---------------------------------------------
'-------------Input data to table--------------
'---------------------------------------------

if ((chindex = 0) or (count = 0))then
  deflt = MsgBox.Input("Is this an open or closed system?  1 = Open/Mixed, 2 =
Closed","User Input","1")
  deflt = deflt.AsNumber

  If (deflt = 1) then
    scheck = 1
    Tstring = "Drain (Unsealed)"
  elseif (deflt = 2) then
    scheck = 2
    TString = "Drain (Sealed)"
  else
    MsgBox.Info("That is not a valid choice","Invalid")
    exit
  end

  For each record in InTab
    InTab.SetValue(DSealFld, record,scheck)
  end

  For each record in TheFtab
    tcheck = TheFtab.ReturnValue(TFld,record)
    if ((tcheck = "Drain (Unsealed)") or (tcheck = "Drain (Sealed)")) then
      TheFtab.SetValue(TFld,record,Tstring)
    else
    end
  end

  labels = {"Diameter (m)", "Temperature (C)"}
```

```
  defaults = {"0.05","20"}
  OutList = MsgBox.MultiInput("Enter the default drain diameter and
temperature","User Input",labels,defaults)
  NewDiam = OutList.Get(0).AsNumber
  NewTemp = OutList.Get(1).AsNumber
  For each record in InTab
    InTab.SetValue(DDiamFld,record,NewDiam)
    InTab.SetValue(DTempFld,record,NewTemp)
  end
else
end

'section to allow changing sealed/unsealed status without losing flow data.
Junctions and manholes also preserved.
If (chIndex = 1) then
  rcheck = MsgBox.YesNoCancel("Change drain status (sealed/unsealed) of all
drains?","User Input",true)
  if (rcheck.AsString = "true") then
    ChoiceList = {"Reverse sealed/unsealed status","Seal all drains","Unseal all
drains"}
    TheChoice = MsgBox.ChoiceAsString(ChoiceList,"Select an option for
changing drain status.","User Input")
    ChoiceIdx = ChoiceList.FindbyValue(TheChoice)
    if (ChoiceIDX = -1) then
      TheFtab.SetEditable(False)
      InTab.SetEditable(False)
      TabDoc = av.FindDoc("drains.dbf")
      TheProject.RemoveDoc(TabDoc)
      TheMainDir.SetCWD
      MsgBox.Info("No changes made to drains","User Input")
      MsgBox.Info("Please click on a drain you would like to edit","User Input")
      exit
    elseif (ChoiceIdx = 0) then
      For each record in TheFtab
        tcheck = TheFtab.ReturnValue(TFld,Record)
        if (tcheck = "Drain (Unsealed)") then
          TheFtab.SetValue(TFld,record,"Drain (Sealed)")
        elseif (tcheck = "Drain (Sealed)") then
          TheFtab.SetValue(TFld,record,"Drain (Unsealed)")
        else
        end
      end
      For each rec in InTab
        scheck = InTab.ReturnValue(DsealFld,rec)
        if (scheck = 1) then
          InTab.SetValue(DsealFld,rec,2)
        else
          InTab.SetValue(DsealFld,rec,1)
        end
      end
    elseif (ChoiceIdx = 1) then
      For each record in TheFtab
        tcheck = TheFtab.ReturnValue(TFld,record)
        If (tcheck = "Drain (Unsealed)") then
          TheFtab.SetValue(Tfld,record,"Drain (Sealed)")
        else
        end
      end
      For each rec in InTab
        scheck = InTab.ReturnValue(DsealFld,rec)
        if (scheck = 1) then
          InTab.SetValue(DsealFld,rec,2)
        else
        end
      end
    else
      For each record in TheFtab
        tcheck = TheFtab.ReturnValue(TFld,record)
        if (tcheck = "Drain (Sealed)") then
          TheFtab.SetValue(TFld,record,"Drain (Unsealed)")
        else
        end
      end
      For each rec in Intab
        scheck = Intab.ReturnValue(DsealFld,rec)
        if (scheck = 2) then
          InTab.SetValue(DsealFld,rec,1)
        else
        end
      end
    end
  else
  end
else
end


TheFtab.SetEditable(False)
```

```
InTab.SetEditable(False)
TabDoc = av.FindDoc("drains.dbf")
TheProject.RemoveDoc(TabDoc)
TheTheme.UpdateLegend

TheMainDir.SetCWD
msgbox.Info("Please click on the drain you would like to edit.","User Input")

’----------------------------------------------
’----------------END SCRIPT--------------------
’----------------------------------------------
```

```
’----------------------------------------------
’---------Creation Information-----------------
’----------------------------------------------
’
’Name: draindata
’Author:  Cindy How
’Date:    6/16/97
’Revisions:
’1.  7/17/97 by Cindy How
’    - Editted to open the correct drains.dbf file
’      (i.e. in the correct directory)
’      Opens, edits, then closes the correct file.
’2.  7/31/97 by Cindy How
’    - Editted to color drain by drain type
’
’----------------------------------------------
’------------Purpose/Description---------------
’----------------------------------------------
’
’This script will take use the node theme and put
’data into the table created by the script
’"draininput".  The user is to click on drains
’which are active (have inflow) and will be prompted
’for necessary information.
’
’Note:  This script may need to change depending
’on if changes are made due to the node/junction
’relationship.  Presently, each node corresponds
’to a drain.  Each "Inode" in the inodes.shp attribute
’table corresponds with a "Isbl_ref" value in the
’drains.dbf table.  Record numbers correspond
’in both tables.
’
’----------------------------------------------
’-------------------Input----------------------
’----------------------------------------------
’
’1.  Theme inodes.shp
’2.  Attribute table of inodes.shp
’3.  Empty table drains.dbf
’
’----------------------------------------------
’------------------Output----------------------
’----------------------------------------------
```

```
'
'1.  Filled table drains.dbf
'
'
'----------------------------------------------
'-----------Get Initial Information-----------
'----------------------------------------------
TheProject = av.GetProject
TheView = av.GetActiveDoc
TheMainDir = FN.GetCWD
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
'Msgbox.info("The new directory is"++TheName.AsString,"test")
TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(3)
TheProject.AddDoc(TheDoc)
TheTheme = TheView.findTheme("inodes.shp")
TheFtab = TheTheme.GetFtab
TheFtab.SetEditable(True)
Tabname = av.GetProject.FindDoc("drains.dbf")
DrainTab = TabName.GetVtab
DrainTab.SetEditable(True)

'----------------------------------------------
'----------------Find Fields------------------
'----------------------------------------------
NFld = TheFtab.FindField("Node")
TypeFld = TheFtab.FindField("Type")


RefFld = DrainTab.FindField("ISBL_ref")
FFld = DrainTab.FindField("Flow_rate")
TFld = DrainTab.FindField("Temperature")
CFld = DrainTab.FindField("Concentrat")
DFld = DrainTab.FindField("Diameter")
SFld = DrainTab.FindField("Sealed")
OFld = DrainTab.FindField("Oil_frac")

'----------------------------------------------
'-----------User Prompted Data Entry-----------
'----------------------------------------------

NewPt = TheView.GetDisplay.ReturnUserPoint
```



```
PtOnTheme = TheTheme.FindByPoint(NewPt)
if (PtOnTheme.isEmpty) then
  Msgbox.error("That point is not found, try again","Not found")
  DrainTab.SetEditable(False)
  TabDoc = av.FindDoc("drains.dbf")
  TheProject.RemoveDoc(TabDoc)
  TheMainDir.SetCWD
  exit
end
rec=PtOnTheme.get(0)
recnum = rec

Inodenum = rec + 1

refList = List.Make
for each record in DrainTab
  Refnum = DrainTab.ReturnValue(reffld,record)
  RefList.Add(RefNum)
end

rec = RefList.FindByValue(INodeNum) 'record number in DrainTab

ChoiceList = {"Change sealed/unsealed","Enter flow data"}
Option = MsgBox.ChoiceAsString(ChoiceList,"Edit which characteristics?","User
Input")
Idx = ChoiceList.FindByValue(option)


If (Idx = 0) then
  seal = DrainTab.ReturnValue(Sfld,rec)
  if (seal = 1) then
    stype = "Open"
  else
    stype = "Closed"
  end
  labels = {"Unsealed drain","Sealed drain"}
  NewSeal = MsgBox.ChoiceAsString(labels,"The drain is presently flagged
as"++stype+".  Indicate the drain type.","User Input")
  ListIdx = labels.FindByValue(NewSeal)
  If (ListIdx < 0) then
    MsgBox.Info("No change made to the drain characteristics","Exit")
    DrainTab.SetEditable(False)
    TabDoc = av.FindDoc("drains.dbf")
    TheProject.RemoveDoc(TabDoc)
```

```
  TheMainDir.SetCWD
  exit
 elseif (ListIdx = 0) then
  DrainTab.SetValue(Sfld,rec,1)
  TheFtab.SetValue(Typefld,recnum,"Drain (Unsealed)")
 else
  DrainTab.SetValue(Sfld,rec,2)
  TheFtab.SetValue(TypeFld,recnum,"Drain (Sealed)")
 end
elseif (Idx = 1) then
 OldFlow = DrainTab.ReturnValue(FFld,rec).AsString
 OldTemp = DrainTab.ReturnValue(TFld,rec).AsString
 OldConc = DrainTab.ReturnValue(CFld,rec).AsString
 OldDiam = DrainTab.ReturnValue(DFld,rec).AsString
 OldOilF = DrainTab.ReturnValue(OFld,rec).AsString
 labels = {"Flow rate (L/s)","Temperature (C)","Concentration
(mg/L)","Diameter (m)","Oil fraction (by volume)"}
 defaults = {oldflow,oldtemp,oldconc,olddiam,oldoilf}
 OutList = MsgBox.MultiInput("Change the values for your selected drain","User
Input",labels,defaults)
 If (OutList.IsEmpty) then
  MsgBox.Info("No changes made to the drain","Cancel")
  DrainTab.SetEditable(False)
  TabDoc = av.FindDoc("drains.dbf")
  TheProject.RemoveDoc(TabDoc)
  TheMainDir.SetCWD
  exit
 else
  NewFlow = OutList.Get(0).AsNumber
  NewTemp = Outlist.Get(1).AsNumber
  NewConc = Outlist.Get(2).AsNumber
  NewDiam = Outlist.Get(3).AsNumber
  NewOilF = OutList.Get(4).AsNumber
  DrainTab.SetValue(FFld,rec,NewFlow)
  DrainTab.SetValue(TFld,rec,NewTemp)
  DrainTab.SetValue(CFld,rec,NewConc)
  DrainTab.SetValue(DFld,rec,NewDiam)
  DrainTab.SetValue(OFld,rec,NewOilF)
 end
else
 MsgBox.Info("No changes made to the drain","Cancel")
end

TheFtab.SetEditable(False)
DrainTab.SetEditable(False)
TabDoc = av.FindDoc("drains.dbf")
TheProject.RemoveDoc(TabDoc)
TheMainDir.SetCWD
TheTheme.UpdateLegend

'---------------------------------------------
'----------------END SCRIPT--------------------
'---------------------------------------------


'---------------------------------
'-------Creation Information-------
'---------------------------------
'
'Name: Iprompt3
'Author: Cindy How
'
'---------------------------------
'-------Purpose/Description--------
'---------------------------------
'
'This script should be run in
'conjunction with the script DROPDATA.
'It will prompt the user to click on
'a branch with which a drop is
'associated.
'


TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme = TheView.FindTheme("ibranch.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.FindField("Drop")
TheFtab.SetEditable(true)


TheMainDir = FN.GetCWD
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(1)
TheProject.AddDoc(TheDoc)
```

```
TabName = TheProject.FindDoc("drops.dbf")
DropTab = TabName.GetVtab
DropTab.SetEditable(true)

flag = 0
If (TheField.AsString = "nil") then
  flag = 1
  TheField = Field.Make("Drop",#Field_char,18,0)
  TheFtab.AddFields({TheField})
else
end

Reset = msgbox.longYesNo("Reset all branches (delete existing drop
data)?","User Input",false)

if ((reset.AsString = "true") or (flag = 1))then
  for each record in TheFtab
    TheFtab.SetValue(TheField,record,"Branch (no drop)")
  end
  For each record in DropTab
    DropTab.RemoveRecord(record)
  end
else
end

TheFtab.SetEditable(False)
TheMainDir.SetCWD
TheProject.RemoveDoc(TheDoc)

TheLegend = TheTheme.GetLegend
TheLegend.Load("sewerlg2.avl".AsFilename,#Legend_loadtype_all)
TheTheme.UpdateLegend



msgbox.Info("Please click on each branch were a drop occurs.  When done, move
on.","User Input")

'----------------------------------
'-----------END SCRIPT-------------
'----------------------------------
```

```
'---------Creation Information-----------------
'
'Name: dropdata
'Author:  Cindy How
'Date:    6/17/97
'
'----------------------------------------------
'------------Purpose/Description---------------
'----------------------------------------------
'
'This program was created from an old version of
'CREATETAB.  It is to be run in conjunction with
'the Iprompt3.  It will add a record to the
'drop.dbf table for each drop and fill in values
'of that table.  The user will be prompted to
'select or accept some default values.
'
'-----------------------------------------------
'-------------------Input-----------------------
'-----------------------------------------------
'
'1.  Empty table drops.dbf
'2.  Theme inodes.shp
'3.  Attribute table of inodes.shp
'4.  Theme ibranch.shp
'5.  Attribute table of ibranch.shp
'
'-----------------------------------------------
'------------------Output-----------------------
'-----------------------------------------------
'
'1.  Filled table called drops.dbf
'
'-----------------------------------------------
'---------Getting initial data------------------
'-----------------------------------------------
theProject=av.getProject
TheView = av.GetActiveDoc
BrTheme = theView.FindTheme("ibranch.shp")
BrTab = BrTheme.GetFtab
NdTheme = TheView.FindTheme("inodes.shp")
NdTab = NdTheme.GetFtab

TheMainDir = FN.GetCWD
```

```
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(1)
TheProject.AddDoc(TheDoc)

TabName=av.GetProject.FindDoc("drops.dbf")
DropTab = TabName.GetVtab
DropTab.SetEditable(True)
'
'-----------------------------------------------
'----------------Find Fields--------------------
'-----------------------------------------------

'InDropTab
NFld = DropTab.FindField("Node")
BFld = DropTab.FindField("Branch")
HFld = DropTab.FindField("Height")
TFld = DropTab.FindField("Tailwater")

'InNdTab
NdFld = NdTab.FindField("Node")
TypeFld = NdTab.FindField("Type")

'InBrTab
BrFld = BrTab.FindField("Ibranch_")
TNdFld = BrTab.FindField("Tnode_")
DrFld = BrTab.FindField("Drop")
BrTab.SetEditable(true)


'------------------------------------------------
'-------------Set Default Values-----------------
'------------------------------------------------
height = 1
depth = 0.5


'------------------------------------------------
'---------------Get Branch Data------------------
'------------------------------------------------
NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = BrTheme.FindByPoint(NewPt)
If (PtOnTheme.IsEmpty) then
  MsgBox.Error("That point is not found, try again","Not found")
  TheProject.RemoveDoc(TheDoc)
  exit
end
rec = PtOnTheme.Get(0)

BrNum = BrTab.ReturnValue(BrFld,rec)
Tnode = BrTab.ReturnValue(TNdFld,rec)
NodeRec = Tnode - 1

'-----------------------------------------------
'---------------Get Node Data-------------------
'-----------------------------------------------

NNodeNum = NdTab.ReturnValue(NdFld,NodeRec)

'----------------------------------------------
'------------Add and fill records--------------
'----------------------------------------------


labels = {"Height (m)","Tailwater depth (m)"}
defaults = {height.AsString,depth.AsString}
OutList = MsgBox.MultiInput("Enter the drop characteristics","User
Input",labels,defaults)

test = OutList.Count

count = 0
For each record in DropTab
  count = count + 1
end

If (test > 0) then
  recnum = DropTab.AddRecord
  Ht = OutList.Get(0).AsNumber
  Twd = Outlist.Get(1).AsNumber
'  if (count > 0) then
'     recnum = count
'   else
'     recnum = 0
```

```
'  end
  DropTab.SetValue(NFld,recnum,NNodeNum)
  DropTab.SetValue(BFld,recnum,BrNum)
  DropTab.SetValue(HFld,recnum,Ht)
  DropTab.SetValue(TFld,recnum,Twd)
  Brtab.SetValue(DrFld,rec,"Branch (drop)")

Else
  MsgBox.Info("No drop added at that location.","No Drop")
end

DropTab.SetEditable(False)
BrTab.SetEditable(False)

TabDoc = av.FindDoc("drops.dbf")
theProject.RemoveDoc(TabDoc)
TheMainDir.SetCWD

TheLegend = BrTheme.GetLegend
TheLegend.Load("sewerlg2.avl".AsFilename,#Legend_loadtype_all)
BrTheme.UpdateLegend

'----------------------------------
'-----------END SCRIPT-------------
'---------------------------------
```

```
'---------------------------------------
'-----------Creation Information---------
'---------------------------------------
'
'Name: Iprompt4
'Author: Cindy How
'
'---------------------------------------
'----------Purpose/Description-----------
'---------------------------------------
'
'This script should be run as the click
'event for the Apply even HPSELECT.  It
'prompts the user to click on a node where
'a hardpipe connection occurs.
'

MsgBox.info("Click on the nodes where hardpipe connections occur.  Move on if
none exist","User Input")

'---------------------------------------
'---------------END SCRIPT---------------
'---------------------------------------
```

```
’---------Creation Information-----------------
’
’Name: hpselect
’Author:  Cindy How
’Date:    6/16/97
’Revisions:
’1. 1/12/98 by Cindy How
’   - Insured only 1 hardpipe connection per node
’     (as per naUTilus requirements)
’
’----------------------------------------------
’------------Purpose/Description---------------
’----------------------------------------------
’
’This program will allow the user to input
’information on hard pipe connections.  The node
’number (by the naUTilus numbering convention)
’will be saved to a new table, as well as informa-
’tion on flowrate, temperature, concentration,
’and oil fraction.
’
’This script is to be run with the script
’"iprompt4".  Hpselect allows the user to click on the node
’at which the hardpipe connection occurs and to
’specify the information needed.
’
’-----------------------------------------------
’-------------------Input-----------------------
’-----------------------------------------------
’
’1.  Node theme (inodes.shp)
’2.  Attribute table of node theme
’3.  Empty table called hardpipe (created by script
’    "hardpipe").
’
’-----------------------------------------------
’------------------Output-----------------------
’-----------------------------------------------
’
’1.  Data filled table called hardpipe
’
’-----------------------------------------------
’---------Getting initial data------------------
’-----------------------------------------------
```

```
theProject=av.getProject
TheView = av.GetActiveDoc
TheTheme = TheView.FindTheme("inodes.shp")
TheFtab = TheTheme.GetFtab
BrTheme = Theview.Findtheme("Ibranch.shp")
BrTab = BrTheme.GetFtab

TheMainDir = FN.GetCWD
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(0)
TheProject.AddDoc(TheDoc)


TabName = av.GetProject.FindDoc("hardpipe.dbf")
HPTab = TabName.GetVtab
HPTab.SetEditable(True)

’-----------------------------------------------
’-----------------Find Fields-------------------
’-----------------------------------------------
InodeFld = TheFtab.Findfield("Inode")
NodeFld = TheFtab.FindField("Node")
TypeFld = TheFtab.FindField("Node_Type")

TNodeFld = BrTab.FindField("Tnode_")

CFld=HPTab.FindField("Connect")
FFld=HPTab.FindField("Flow_rate")
TFld=HPTab.FindField("Temp")
ConFld = HPTab.FindField("Conc")
OilFld = HPTab.FindField("Oil_fraction")

’-----------------------------------------------
’----------------Get Point----------------------
’-----------------------------------------------
NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = TheTheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
     Msgbox.error("That point is not found, try again","Not found")
     HPTab.SetEditable(False)
     TabDoc = av.FindDoc("hardpipe.dbf")
```

```
    TheProject.RemoveDoc(TabDoc)
    TheMainDir.SetCWD
    exit
  end
rec=PtOnTheme.get(0)


TheNode = TheFtab.ReturnValue(InodeFld,rec)
NNode = TheFtab.ReturnVAlue(NodeFld,rec)

TheCount = 0
For each record in HPTab
  TheCNode = HPTab.ReturnValue(CFld,record)
  if (TheCNode = NNode) then
    TheCount = 1
    Edcheck = msgbox.LongYesNo("Hardpipe connection already exists at the
node.  Edit?","User Input",true)
    If (EdCheck.AsString = "true") then
      Flow = HPTab.ReturnValue(FFld,record).AsString 'get old values
      Temp = HPTab.ReturnValue(TFld,record).AsString
      Conc = HPTab.ReturnValue(ConFld,record).AsString
      Frac = HPTAb.ReturnValue(OilFld,record).AsString
'user input new values
      defaults = {Flow,Temp,Conc,Frac}
      labels = {"Flow rate (L/s)","Temperature (C)","Concentration (mg/L)","Oil
Fraction by volume"}
      OutList = MsgBox.MultiInput("Enter hardpipe connection
characteristics","User Input",labels,defaults)
      Flow = Outlist.Get(0).AsNumber
      Temp = Outlist.Get(1).AsNumber
      Conc = Outlist.Get(2).AsNumber
      Frac = Outlist.Get(3).AsNumber
'set new values in table
      HPTab.SetValue(FFld,record,flow)
      HPTab.SetValue(TFld,record,temp)
      HPTab.SetValue(ConFld,record,conc)
      HPTab.SetValue(OilFld,record,Frac)
    else
      MsgBox.Info("No hardpipe connection added","Info")
      HPTab.SetEditable(False)
      TabDoc = av.FindDoc("hardpipe.dbf")
      TheProject.RemoveDoc(TabDoc)
      TheMainDir.SetCWD
      exit
    end
  else
  end
end

If (TheCount = 0) then 'Adding New Hardpipe Connection
  NewRec = HPTab.AddRecord
  defaults = {".1","30","1","0.0"}
  labels = {"Flow rate (L/s)","Temperature (C)","Concentration (mg/L)","Oil
Fraction by volume"}
  OutList = MsgBox.MultiInput("Enter hardpipe connection characteristics","User
Input",labels,defaults)
  check2=Outlist.Count
  if (Check2 = 0) then 'cancelled addition of hardpipe connection
    HPTab.SetEditable(False)
    TabDoc = av.FindDoc("hardpipe.dbf")
    TheProject.RemoveDoc(TabDoc)
    TheMainDir.SetCWD
    exit
  else
    NodeNum = TheFtab.ReturnValue(NodeFld,rec)
    Flow = Outlist.Get(0).AsNumber
    Temp = Outlist.Get(1).AsNumber
    Conc = Outlist.Get(2).AsNumber
    Frac = Outlist.Get(3).AsNumber

    HPTab.SetValue(CFld,NewRec,NodeNum)
    HPTab.SetValue(FFld,NewRec,Flow)
    HPTab.SetValue(TFld,NewRec,Temp)
    HPTab.SetValue(ConFld,NewRec,Conc)
    HPTab.SetValue(OilFld,NewRec,Frac)
  end
else
end

HPTab.SetEditable(False)

TabDoc = av.FindDoc("hardpipe.dbf")
TheProject.RemoveDoc(TabDoc)
TheMainDir.SetCWD

'----------------------------------------
'---------------END SCRIPT---------------
'----------------------------------------
```

```
'---------------------------------------------
'-------------Creation Information------------
'---------------------------------------------
'
'Name: Iprompt5
'Author:  Cindy How
'Date:  7/17/97
'
'---------------------------------------------
'-----------Purpose/Description---------------
'---------------------------------------------

'This script will let the user reset branch
'characteristics with different options.

'If they accept the default (1) characteristics, diameters for
'branches flowing from junctions is 0.15337 and for those
'flowing from manholes is 0.20552.
'
'The default (2) option is to set the all branch diameters equal
'to a user selected value.
'
'For both these options, individual branch diameters can be
'changed by the user.  The slope will default originally be
'0.01, a value which can be changed for individual branches
'or to a new default by the user.

ChList = {"Reset to Default (1)","Reset to Default (2)","Edit individual
branches"}
TheChoice = MsgBox.ChoiceAsString(ChList,"Select a mode: naUTilus
defaults(1), user defaults(2), edit branches","User Input")
ChIdx = ChList.FindByValue(TheChoice)


If ((ChIdx = 0) or (ChIdx = 1)) then
  TheProject = av.GetProject
  TheView = av.GetActiveDoc
  TheTheme = TheView.FindTheme("ibranch.shp")
  TheFtab = TheTheme.GetFtab
  FNdFld = TheFtab.FindField("Fnode_")

  NdTheme = TheView.FindTheme("inodes.shp")
  NdTab = NdTheme.GetFtab
  MHFld = NdTab.FindField("Manhole")
  AVNdFld = NdTab.FindField("Inode")

  TheMainDir = FN.GetCWD
  TheWin = TheView.GetWin
  TheName = TheWin.GetTitle.AsFilename
  TheDir = TheName.SetCWD
  TheODB = ODB.Open("isbl.odb".AsFilename)
  TheDoc = TheODB.Get(2)
  TheProject.AddDoc(TheDoc)


  TheTabName = av.GetProject.FindDoc("brtable.dbf")
  TheTab = TheTabName.GetVtab
  DiamFld = Thetab.Findfield("Diameter")
  SlFld = Thetab.FindField("Slope")
  EDLFld = TheTab.FindField("Length")
  TheTab.SetEditable(true)

'find editted length field (if drawing is not to scale and field has been created)

  TheEDLFld = TheFtab.FindField("Length_ed")
  If (TheEdlFld.AsString = "nil") then
  'no action
  else
    For each rec in TheFtab
      EDLen = TheFtab.ReturnValue(TheEdlFld,rec)
      TheTab.SetValue(EDLFld,rec,EDLen)
    end
  end

  If (ChIdx = 0) then
    MHList = List.Make
    For each record in NdTab
      MHcheck = NdTab.ReturnValue(MHFld,record)
      If (MHcheck = 1) then
        AVnode = NdTab.ReturnValue(AVNdFld,record)
        MHList.Add(AVnode)
      else
      end
    end

    For each record in TheFtab
      FNd = TheFtab.ReturnValue(FNdFld,record)
      check = MHList.FindbyValue(FNd)
```

```
    If (check > -1) then
      TheTab.SetValue(DiamFld,record,0.20552)
    else
    end
  end
else
  NewDiam = msgbox.Input("Enter the default diameter (m)","User
Input","0.15337")
  if ((NewDiam = "nil") or (NewDiam.AsNumber < 0)) then
    MsgBox.Info("Error, no changes made","Not Valid")
    TheTab.SetEditable(false)
    TheTab.SetEditable(false)
    TabDoc = av.FindDoc("brtable.dbf")
    TheProject.RemoveDoc(TabDoc)
    TheMainDir.SetCWD
    exit
  else
    For each record in TheTab
      TheTab.SetValue(DiamFld,record,NewDiam.AsNumber)
    end
  end
end
DefSlope = MsgBox.Input("Enter the default slope (%)","User Input","0.01")
For each record in TheTab
  TheTab.SetValue(SlFld,record,DefSlope.AsNumber)
end
TheTab.SetEditable(false)
TabDoc = av.FindDoc("brtable.dbf")
TheProject.RemoveDoc(TabDoc)
TheMainDir.SetCWD
msgbox.info("Please click on a branch which you wish to edit.  If none, move
on.","User Input")
elseif (ChIdx = -1) then
  exit
else
  msgbox.info("Please click on the branch which you wish to edit.","User Input")
end

'---------------------------------------------
'-----------------END SCRIPT------------------
'---------------------------------------------
```



```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: selectbr
'Author:  Cindy How
'Date:    6/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take select the branch to be
'editted.  It is to be used as the apply event
'with iprompt5.
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  "Brtab.dbf" file
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
'1.  Selected record
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme = TheView.FindTheme("ibranch.shp")

TheMainDir = FN.GetCWD
TheWin = TheView.GetWin
TheName = TheWin.GetTitle.AsFilename
TheDir = TheName.SetCWD
TheODB = ODB.Open("isbl.odb".AsFilename)
TheDoc = TheODB.Get(2)
TheProject.AddDoc(TheDoc)
```

```
TheTabName = av.GetProject.FindDoc("brtable.dbf")
TheTab = TheTabName.GetVtab
DiamFld = Thetab.Findfield("Diameter")
SlopeFld = Thetab.FindField("Slope")
TheTab.SetEditable(true)

NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Thetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
    Msgbox.error("That point is not found, try again","Not found")
    exit
  end
rec=PtOnTheme.get(0)


OldDia = TheTab.ReturnValue(DiamFld,rec)
OldSlope = TheTab.ReturnValue(SlopeFld,rec)
labels = {"Diameter (m)","Slope (%)"}
defaults = {OldDia.AsString,OldSlope.AsString}
NewValList = msgbox.multiInput("Enter the branch characteristics","New
Values",labels,defaults)
test = NewValList.Count
If (test > 0) then
  NewDiam = NewValList.get(0).AsNumber
  NewSlope = NewValList.Get(1).AsNumber
  TheTab.SetValue(DiamFld,rec,NewDiam)
  TheTab.SetValue(SlopeFld,rec,NewSlope)
else
end

TheTab.SetEditable(false)
TabDoc = av.FindDoc("brtable.dbf")
TheProject.RemoveDoc(TabDoc)
TheMainDir.SetCWD

'---------------------------------------------
'-----------------END SCRIPT------------------
'---------------------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: writeinp
'Author:  Cindy How
'Date:    6/97
'Revisions:
'1.  7/15/97 by Cindy How
'    - Added lines to delete old versions of
'      script created tables and open new ones
'      found through ODB file.
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take the .dbf files created
'in previous steps and create the naUTilus input
'file.  The input file will be
'called "isbl.in"
'
'This script assumes that the ISBL unit for which
'an input file is being created is the open, active
'window (View window).
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  Node theme
'2.  Attribute table of the Node theme
'3.  Branch theme
'4.  Attribute table of the Branch theme
'5.  drains.dbf table
'6.  nodecon.dbf table
'7.  brtable.dbf table
'8.  drops.dbf table
'9.  hardpipe.dbf table
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
```

```
'1.  Text input file, "isbl.in"
'
'
'-----------------------------------------------
'------------------Notes----------------------
'-----------------------------------------------
'
'The variables InList, InDefault, and OutList are
'temporary and used at several points in the script.
'They take on new values several times.
'
'-----------------------------------------------
'---------Get Initial Information---------------
'-----------------------------------------------
TheMainDir = FN.GetCWD 'change this later

TheProject = av.GetProject
TheView = av.GetActiveDoc


ViewWin = TheView.GetWin
TheViewName = ViewWin.GetTitle.AsString
av.run("connctit",{TheViewName})


ViewWin.Activate

TheISBL = ViewWin.GetTitle.AsFilename

TheISBL.SetCWD


NdTheme = theview.Findtheme("inodes.shp")
BrTheme = TheView.Findtheme("ibranch.shp")
NodeTab = NdTheme.GetFtab
BranchTab = BrTheme.GetFtab

DTabName = av.FindDoc("drains.dbf")
CTabName = av.FindDoc("nodecon.dbf")
BTabName = av.Finddoc("brtable.dbf")
DropName = av.FindDoc("drops.dbf")
HPName = av.FindDoc("hardpipe.dbf")

'remove old versions of the tables (and check for existing)

If (DTabName.AsString = "nil") then
else
  TheProject.RemoveDoc(DtabName)
end

If (CTabName.AsString = "nil") then
else
  TheProject.RemoveDoc(CTabName)
end

If (BTabName.AsString = "nil") then
else
  TheProject.RemoveDoc(BTabname)
end

If (DropName.AsString = "nil") then
else
  TheProject.Removedoc(Dropname)
end

If (HPName.AsString = "nil") then
else
  TheProject.RemoveDoc(HPName)
end

'Add correct versions of the tables from ODB
TheODB = ODB.Open("isbl.odb".AsFilename)

Doc0 = TheODB.Get(0)
Doc1 = TheODB.Get(1)
Doc2 = TheODB.Get(2)
Doc3 = TheODB.Get(3)
Doc4 = TheODB.Get(4)

TheProject.AddDoc(Doc0)
TheProject.AddDoc(Doc1)
TheProject.AddDoc(Doc2)
TheProject.AddDoc(Doc3)
TheProject.AddDoc(Doc4)

DTabName = av.FindDoc("drains.dbf")
DrainTab = DTabName.GetVtab
CTabName = av.FindDoc("nodecon.dbf")
ConnTab = CTabName.GetVtab
```

```
BTabName = av.Finddoc("brtable.dbf")
BrTab = BTabName.GetVtab
DropName = av.FindDoc("drops.dbf")
DropTab = DropName.GetVtab
HPName = av.FindDoc("hardpipe.dbf")
HPTab = HPName.GetVtab



'-----------------------------------------------
'------------------Find Fields------------------
'-----------------------------------------------
'In BrTab
LFld = BrTab.Findfield("Length")
DFld = BrTab.Findfield("Diameter")
SFld = BrTab.FindField("Slope")

'In DrainTab
DtypeFld = DrainTab.FindField("Drain_type")
ConFld = DrainTab.FindField("Connectivi")
FlFld = DrainTab.FindField("Flow_rate")
TempFld = DrainTab.Findfield("Temperatur")
CFld = DrainTab.FindField("Concentrat")
DDFld = DrainTab.FindField("Diameter")
SealFld = DrainTab.Findfield("Sealed")
OilFld = DrainTab.FindField("Oil_frac")

'InConnTab
Fld1 = ConnTab.FindField("Cola")
Fld2 = ConnTab.FindField("Colb")

'InNodeTab
NNumFld = NodeTab.Findfield("Node")
ArcFld = NodeTab.Findfield("Arc_")
TypeFld = NodeTab.Findfield("Node_Type")

'InBranchTab
FFld = BranchTab.Findfield("Fnode_")

'InDropTab
DNdFld = DropTab.FindField("Node")
DBrFld = DropTab.Findfield("Branch")
DHtFld = DropTab.FindField("Height")
DTWDFld = DropTab.FindField("Tailwater")
```



```
'InHPTab
HPCFld = HPTab.FindField("Connect")
HPFFld = HPTab.FindField("Flow_rate")
HPTFld = HPTab.FindField("Temp")
HPConFld = HPTab.FindField("Conc")
HPOilFld = HPTab.FindField("Oil_fracti")

'----------------------------------------------
'----------------Make Text File-----------------
'----------------------------------------------

NewFile = "isbl.in".AsFileName
LFile = LineFile.Make(NewFile,#File_Perm_write)

Lfile.WriteElt("no.branches, no.nodes, no. drains, open/closed")
LFile.WriteElt("1 = open/mixed, 2 = closed")

numbr = 0
For each record in Brtab
  numbr = numbr + 1
end

numnode = 0
For each record in Nodetab
  Nd = NodeTab.ReturnValue(NNumFld,record)
  if (nd < 1) then
  else
    numnode = numnode + 1
  end
end

numdr = 0
sealcheck = 2
for each record in drainTab
  numdr = numdr + 1
  sealflag = DrainTab.ReturnValue(SealFld,record)
  if (sealflag < 2) then
    sealcheck = 1
  else
  end
end
```

```
Lfile.WriteElt(numbr.AsString+","+numnode.AsString+","+numdr.AsString+","+
sealcheck.AsString)

Lfile.WriteElt("last node")

'finds the last node in the system by the naUTilus numbering system
arclist = list.make
TypeList = list.make
For each record in NodeTab
  arcnum = NodeTab.ReturnValue(ArcFld,record)
  ntype = NodeTab.ReturnValue(TypeFld,record)
  ArcList.Add(arcnum)
  TypeList.Add(ntype)
end
IdxNum = TypeList.FindByValue(2)
arcid = ArcList.Get(IdxNum)
arcidx = arcid - 1
LNodeAv = BranchTab.ReturnValue(FFld,arcidx)
LNodeAvIdx = LNodeAv - 1
LastNode = NodeTab.ReturnValue(NNumFld,LNodeAvIdx)

LFile.WriteElt(LastNode.AsString)

'prompt user to input Henry's law method

MethodList = {"Hc=exp(A-B/T)","Hc at known T, adjusted using Antoine's
constants","VP/Sol using Antoine's constants","Hc=constant"}
'MethodList = {"1","2","3","4"}
meth = Msgbox.ChoiceAsString(MethodList,"Select the method used to
determine Henry's constant:","Henry's law Constant")
MethodIdx = MethodList.FindByValue(meth)
Method = MethodIdx + 1

LFile.WriteElt("Henry's law Constant - Method 1,2,3,4")
'TAG! in ISBL code, remove dummy variables (9 lines) before running!
LFile.WriteElt(Method.AsString)

'TAG! Find better default values for defaults below!
'Prompt for correct variables (dependent on Method)
If (Method = 1) then
  InLabels = {"A","B"}
  InDefaults = {"5.524","3194.0"}
  OutList = Msgbox.MultiInput("Enter the values of A and B (Hc = exp(A-
B/T))","Method 1",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then
    LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
    TheMainDir.SetCWD
    av.run("remdoc",{})
  exit
  end
elseif (Method = 2) then
  InLabels = {"Hc at known temperature","T in C (known
temperature)","Antoine's A","Antoine's B","Antoine's C"}
  InDefaults = {"0.5","25","1","1","1"}
  Outlist = Msgbox.MultiInput("Enter the values (Antoine constants for T in K,
giving vapor Presure in mmHg","Method 2",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString+","+Outlist.Get(3).AsString+","+Outlist.Get(4).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
    TheMainDir.SetCWD
    av.run("remdoc",{})
  exit
  end
elseif (Method = 3) then
  InLabels = {"A","B","C","Solubility (g/L)","Molecular Weight (g/mol)"}
  InDefaults = {"1","1","1","1","100"}
  OutList = MsgBox.MultiInput("Enter the values of Antoine's A, B, and C,
solubility, and MW","Method 3",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString+","+Outlist.Get(3).AsString+","+Outlist.Get(4).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
    TheMainDir.SetCWD
    av.run("remdoc",{})
  exit
```

```
  end
elseif (Method = 4) then
  InLabels = {"Hc"}
  InDefaults = {"0.5"}
  OutList = MsgBox.MultiInput("Enter the Henry’s Constant","Method
4",Inlabels,InDefaults)
  check = OutList.Count
  If (check > 0) then
    LFile.WriteElt(Outlist.Get(0).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
    TheMainDir.SetCWD
    av.run("remdoc",{})
  exit
  end
else
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
end

’User Prompt for diffusivities
InList = {"Liquid Phase Diffusivity (cm2/s)","Gas Phase Diffusivity (cm2/s)"}
InDefault = {"1.0e-5","0.1"}
outList = MsgBox.MultiInput("Enter the Diffusivities","User
Input",InList,InDefault)
’check for cancel
test = Outlist.count
if (test = 0) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
  ’write diffusivities
  Lfile.WriteElt("Liquid Diffusivity, Gas Diffusivity")
  LFile.WriteElt(OutList.Get(0).AsString+","+OutList.Get(1).AsString)
end

’User prompt for ambient conditions
InList = {"Ambient Temperature (C)","Relative humidity"}
InDefault = {"25","0.5"}
OutList = MsgBox.MultiInput("Enter the Ambient Conditions","User
Input",InList,InDefault)
’check for cancel
test = Outlist.count
if (test = 0) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
    ’write ambient conditions
  LFile.WriteElt("Ambient Temperature (C) and relative humidity")
  LFile.WriteElt(OutList.Get(0).AsString+","+OutList.Get(1).AsString)
end

’User prompt for manhole characteristics
InList = {"No. of pick-holes per cover","Pick-hole diameter (m)"}
InDefault = {"4","0.0254"}
OutList = MsgBox.MultiInput("Enter manhole characteristics","User
Input",InList,InDefault)
’check for cancel
test = Outlist.count
if (test = 0) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
  ’write manhole characteristics
  LFile.WriteElt("no. pickholes per cover, pickhole diameter (m)")
  LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString)
end

’User Prompt for wind speed
LFile.WriteElt("Ambient wind velocity (m/s)")
wspeed = MsgBox.Input("Enter the ambient wind speed (m/s)","User Input","1")

’check for cancel
if (wspeed.AsString = "nil") then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
```

```
'write wind speed
LFile.WriteElt(wspeed.AsString)
end

'User prompt for presence of oil
oilList = {"No oil present","Oil present only at surface","Oil present and dispersed
(not at surface)"}
oilcheck = MsgBox.ChoiceAsString(OilList,"Indicate presence and condition of
oil","User Input").AsString
oilIdx = OilList.FindbyValue(oilcheck)
'check for cancel
if (oilIdx = -1) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
end

oilcond = oilidx + 1
LFile.WriteElt("1 = no oil, 2 = oil only at surface")
LFile.WriteElt("3 = oil only dispersed, not at surface")
LFile.WriteElt(oilcond.AsString)
LFile.WriteElt("If oil is present, read Kow, density, oil MW (else, empty)")

if (oilcond > 1) then
  InLabels = {"Kow","Oil density (g/m3)","Oil molecular weight (g/mol)"}
  InDefaults = {".5","20","100"}
  OutList = MsgBox.MultiInput("Enter oil characteristics","User
Input",InLabels,InDefaults)

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString)
else
end


'fill dummy variable lines (5 of them)
LFile.WriteElt("Indicate method of determining vapor pressure if oil is present")
LFile.WriteElt("Method 1-Antoine's constants")
LFile.WriteElt("Enter A,B,C for T in K giving VP in mmHg")
LFile.WriteElt("Method 2-Vapor Pressure = constant")
LFile.WriteElt("Enter Vapor Pressure")
```



```
if (oilcond > 1) then
  MethList = {"Antoine's constants","Constant vapor pressure"}
  oilmeth = msgbox.ChoiceAsString(MethList,"Indicate method of calculating
vapor pressure","User Input").AsString
  methind = MethList.findByValue(oilMeth)
  indnum = methind + 1
  if (indnum = 1) then
    InLabels = {"A","B","C"}
    InDefaults = {"1","1","1"}
    Outlist = MsgBox.MultiInput("Enter the Antoine's constants (for T in K, giving
vapor pressure in mmHg)","User Input",InLabels,InDefaults)

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString)
  else
    VapPres = MsgBox.Input("Enter the vapor pressure (atm)","User Input","1")
    Lfile.WriteElt(VapPres.AsString)
  end
else
end

'user input - indicate P&P or OEG
LFile.WriteElt("1 = Parkhurst&Pomeroy, 2 = Owens-Edwards-Gibbs")
ChList = {"Parkhurst & Pomeroy","Owens-Edwards-Gibbs"}
ch = Msgbox.ChoiceAsString(ChList,"Calculate using:","User Input").AsString
ChInd = ChList.FindByValue(ch)

'check for cancel
if (ChInd = -1) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
end

Chnum = chInd + 1
LFile.WriteElt(Chnum.AsString)

'User input - indicate calculation: masstransfer(1)/equilibrium(2)
LFile.WriteElt("1 = Mass Transfer, 2 = Equilibrium")
CalcList = {"Mass Transfer","Equilibrium"}
calc = MsgBox.ChoiceAsString(CalcList,"Choose the mechanism","User Input")
CalcInd = CalcList.FindByValue(calc)
```

```
'check for cancel
if (CalcInd = -1) then
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  TheMainDir.SetCWD
  av.run("remdoc",{})
  exit
else
end

calcnum = calcind + 1
LFile.WriteElt(calcnum.AsString)

'do not use default reach length, enter length, diameter, and slope for each reach
'may need to change this later (6/11/97)
LFile.WriteElt("Enter 1 to accept default reach length")
LFile.WriteElt("2")
LFile.WriteElt("User specified branch characteristics")
Lfile.Writeelt("Length, Diameter, slope for each branch")

for each record in BrTab
  Len = BrTab.ReturnValue(LFld,record)
  Diam = BrTab.ReturnValue(DFld,record)
  Slope = BrTab.ReturnValue(SFld,record)
  LFile.WriteELT(len.AsString+","+Diam.AsString+","+Slope.AsString)
end

LFile.WriteElt("Enter drain characteristics: type - elbow(1)/on line(2),
branch/node connectivity, flowrate (L/s),")
LFile.WriteElt("temperature(C), concentration (mg/L), drain diameter (m),")
LFile.WriteElt("unseald(1)/sealed(2> drain, oil vol./total vol.")

for each record in Draintab
  type = Draintab.ReturnValue(DtypeFld,record).AsString
  conn = DrainTab.ReturnValue(ConFld,record).AsString
  rate = DrainTab.ReturnValue(FlFld,record).AsString
  temp = DrainTab.ReturnValue(TempFld,record).AsString
  Conc = DrainTab.ReturnValue(CFld,record).AsString
  Diam = Draintab.ReturnValue(DDFld,record).AsString
  Seal = DrainTab.ReturnValue(SealFld,record).AsString
  Oil = Draintab.ReturnValue(OilFld,record).AsString

LFile.WriteElt(type+","+conn+","+rate+","+temp+","+conc+","+diam+","+seal+"
,"+oil)
end

'Fill dummy variables (4 lines)
LFile.WriteElt("Enter no. of branches connected to node i, kind of node
(junction/manhole)")
LFile.WriteElt("...1 for junction, 2 for manhole...")
LFile.WriteElt("Enter branch no., kind of connection")
LFile.WriteElt("...1 for inflow, -1 for outflow")

for each record in ConnTab
  columna = ConnTab.ReturnValue(Fld1,record).AsString
  columnb = ConnTab.ReturnValue(Fld2,record).AsString
  LFile.WriteElt(columna+","+columnb)
end

numdrops = 0
For each record in Droptab
  numdrops = numdrops + 1
end
LFile.WriteElt("Number of drops")
LFile.WriteElt(numdrops.AsString)

LFile.writeElt("enter node, branch, drop height (m), tailwater depth (m) for each
drop")
If (numdrops = 0) then
else
  For each record in DropTab
    DrNode = DropTab.ReturnValue(DNdFld,record)
    DrBranch = DropTab.ReturnValue(DBrFld,record)
    DrHeight = DropTab.ReturnValue(DHtFld,record)
    DrTWD = DropTab.ReturnValue(DTWDFld,record)

LFile.WriteElt(DrNode.AsString+","+DrBranch.AsString+","+DrHeight.AsString
+","+DrTWD.AsString)
  end
end


LFile.WriteElt("Enter no. of hard-piped connections")

HPcount = 0
For each record in HPTab
  HPCount = HPCount + 1
end
```

```
LFile.WriteElt(HPCount.AsString)
LFile.WriteElt("Enter node connectivity, flow rate (L/s), temperature (C),")
LFile.WriteElt("concentration (mg/L), oil volume/total volume")
If (HPCount > 0) then
  For each record in HPTab
    HPNode = HPTab.ReturnValue(HPCFld,record).AsString
    HPFlow = HPTab.ReturnValue(HPFFld,record).AsString
    HPTemp = HPTab.ReturnValue(HPTFld,record).AsString
    HPConc = HPTab.ReturnValue(HPConFld,record).AsString
    HPFrac = HPTab.ReturnValue(HPOilFld,record).AsString

LFile.WriteElt(HPNode+","+HPFlow+","+HPTemp+","+HPConc+","+HPFrac)
  end
else
end

TheProject.Removedoc(Doc0)
TheProject.RemoveDoc(Doc1)
TheProject.RemoveDoc(doc2)
TheProject.RemoveDoc(doc3)
TheProject.RemoveDoc(Doc4)

TheMainDir.SetCWD

'----------------------------------------------
'----------------END SCRIPT--------------------
'----------------------------------------------
```



```
'----------------------------------------------
'------------Creation Information---------------
'----------------------------------------------
'
'Name: ISBLBAT
'Creator: Cindy How
'Date:  8/13/97
'
'----------------------------------------------
'-------------------Purpose---------------------
'----------------------------------------------
'
'This script will create a batch file to find the
'correct isbl.in file, copy it to the main directory,
'run naUTilus (ISBL), copy the output back to the
'correct subdirectory, and delete the input and
'output files from the main directory.
'
'----------------------------------------------
'-------------Get Initial Information-----------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheWin = TheView.GetWin
ISBLName = TheWin.GetTitle

TheFName = "isbl.bat".AsFileName
LFile = LineFile.Make(TheFName,#File_Perm_write)

LFile.WriteElt("copy"++ISBLName+"\isbl.in isbl.in")
LFile.WriteElt("isbl.exe")
LFile.WriteElt("copy isbl.out"++ISBLName)
LFile.WriteElt("copy isblout.txt"++ISBLName)
LFile.WriteElt("del isbl.out")
LFile.WriteElt("del isblout.txt")
LFile.WriteElt("del isbl.in")

av.DelayedRun("irunbat",nil,10)
```

```
'-------------------------------------
'---------Creation Information--------
'-------------------------------------
'
'Name: Irunbat
'Author: Cindy How
'Date: 8/13/97
'
'-------------------------------------
'--------Purpose/Description----------
'-------------------------------------
'
'This script was created to run the
'batch file created by the script
'ISBLBAT.  It was created to enable
'a delayed run of the ISBL module
'of naUTilus.
'
'The run has been designed to be delayed
'due to issues with memory allotment which
'could not otherwise be resolved.


system.execute("isbl.bat")

'-------------------------------------
'-----------END SCRIPT----------------
'-------------------------------------

'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: junction
'Author:  Cindy How
'Date:    6/23/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take all points in the onodes.shp
'theme and find which are terminal points (elbows
'vs online drains or manholes).  It also flags
'the nodes which are junctions with no manholes
'by calling on the script SELJUNC.
'
'FOR USE ON OSBL UNITS
'
'----------------------------------------------
'------------------Input-----------------------
'----------------------------------------------
'
'1.  Attribute table for the theme nodes
'
'----------------------------------------------
'-----------------Output-----------------------
'----------------------------------------------
'
'1.  Added field to node theme, indicating elbow (-1)
'    or online drain or manhole (numbered)
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme= TheView.FindTheme("Onode.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.FindField("Osbl_")
TheField.SetAlias("Onodes")


BrTheme = TheView.FindTheme("Obranch.shp")
BrTab = BrTheme.GetFtab
TNodeField=BrTab.FindField("Tnode_")
FnodeField = BrTab.FindField("Fnode_")
'----------------------------------------------
'-----Add New Fields to Node Attribute Table----
'----------------------------------------------

TheFtab.setEditable(true)
FldList = List.Make
TypeFld = TheFtab.FindField("Node_Type")
If (TypeFld.AsString = "nil") then
  TypeFld=Field.make("Node_Type",#Field_short,4,0)
  FldList.Add(TypeFld)
```

```
else
end


NodeFld = TheFtab.FindField("Node")
If (NodeFld.AsString = "nil") then
  NodeFld=Field.make("Node",#Field_short,4,0)
  FldList.Add(NodeFld)
else
end

NTypeFld = TheFtab.FindField("Junction")
if (NTypeFld.AsString = "nil") then
  NTypeFld = Field.make("Junction",#Field_short,4,0)
  FldList.Add(NTypeFld)
else
end

BFld = TheFtab.FindField("Ex_br")
if (BFld.AsString = "nil") then
  BFld = Field.make("Ex_br",#Field_short,4,0)
  FldList.Add(BFld)
else
end

NLabelFld = TheFtab.FindField("Nd_label")
if (NlabelFld.AsString = "nil") then
  NlabelFld = Field.Make("Nd_label",#Field_char,25,0)
  FldList.Add(NlabelFld)
else
end

TheFtab.AddFields(FldList)


TNodeList = List.Make 'Makes List of "to Nodes"
Fnodelist = list.make 'makes a list of "from nodes"
i=0
For each rec in BrTab
  TNode = BrTab.ReturnValue(TNodeField,i)
  TNodeList.Add(Tnode)
  Fnode = BrTab.ReturnValue(FnodeField,i)
  Fnodelist.Add(Fnode)
    i = i +1
end

j= 1  'Assigns value of -1 to the field "Node_Type" for terminal nodes
For each rec in TheFtab
  Check = TNodeList.FindbyValue(j)
      If (check = -1) then
      TheFtab.SetValue(TypeFld,rec,check)
    else
      TheFtab.SetValue(TypeFld,rec,1)
    end
  Check2 = FnodeList.FindByValue(j)
    If (check2 = -1) then
    TheFtab.SetValue(typeFld,rec,2)
'set type value of last node in system to 2 (last node by ArcView® system)
    Else
    end
    j = j+1
end


'---------------------------------------------
'-----------Junction or Manhole----------------
'---------------------------------------------

'Reset all manholes?
check = MsgBox.LongYesNo("Would you like to
                 reset all junctions and node types?","User Input",true)
if (check.AsString = "true") then
  For each record in TheFtab
    TheFtab.SetValue(NTypeFld,record,0)
'    av.run("graphic.SelectAll",{})
'    av.run("View.DeleteGraphics",{})
    flag = TheFtab.ReturnValue(TypeFld,record)
    if (flag = 5) then
      TheFtab.SetValue(TypeFld,record,-1)
    else
    end
  end
  NBrTab = TheProject.FindDoc("obrtab.dbf")
  If (NbrTab.AsString = "nil") then
  else
    TheProject.RemoveDoc(NbrTab)
  end
else
```

```
end

for each record in TheFtab
  TheType = TheFtab.ReturnValue(TypeFld,record)
  if (TheType = -1) then
    TheFtab.SetValue(NLabelFld,record,"Node (no manhole)")
  elseif (TheType = 1) then
    TheFtab.SetValue(NLabelFld,record,"Node (manhole)")
  elseif (TheType = 5) then
    TheFtab.SetValue(NlabelFld,record,"Junction (no manhole)")
  elseif (TheType = 2) then
    TheFtab.SetValue(NlabelFld,record,"Outlet")
  else
  end
end

TheLegend = TheTheme.GetLegend
TheLegend.Load("osblnd.avl".AsFilename,#Legend_loadtype_all)
TheTheme.UpdateLegend

'Prompt user to select which nodes represent manholes.
'Assign a value of 1 to manholes and -1 to junctions

TheFtab.SetEditable(false)

msgbox.info("Click on a nodes to edit the type","Select node")
'----------------------------------------------
'---------------END SCRIPT---------------------
'----------------------------------------------
```



```
'----------------------------------------------
'---------Creation Information-----------------
'----------------------------------------------
'
'Name: seljunc
'Author:  Cindy How
'Date:    6/23/97
'
'----------------------------------------------
'------------Purpose/Description---------------
'----------------------------------------------
'
'This script will take select the point to be
'flagged as a junction with no manhole.  It is
'run in conjunction with the script JUNCTION, with
'JUNCTION as the Click event and SELJUNC as the
'Apply event.
'
'FOR USE WITH OSBL UNITS
'
'----------------------------------------------
'------------------Input-----------------------
'----------------------------------------------
'
'1.  Node theme
'2.  Attribute table of the Node theme
'
'----------------------------------------------
'-----------------Output-----------------------
'----------------------------------------------
'
'1.  Selected record
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme= TheView.FindTheme("onode.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.Findfield("Junction")
```

```
ShpFld = TheFtab.FindField("Shape")
LabelFld = TheFtab.FindField("Nd_label")
TypeFld = TheFtab.FindField("Node_Type")

TheFtab.SetEditable(true)

NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Thetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
    Msgbox.error("That point is not found, try again","Not found")
    TheFtab.SetEditable(false)
    exit
  end
rec=PtOnTheme.get(0)

check = TheFtab.ReturnValue(TheField,rec)

if (check = 1) then
  check2 = msgbox.longyesno("That node is already
                 flagged as a junction (no manhole).  Reset node?","User Input",true)
  if (check2.asString = "true") then
’   TheGraphicList = TheView.GetGraphics
’   TheGraphic = TheGraphicList.FindByLocation(NewPt)
’   TheGraphicList.RemoveGraphic(TheGraphic)
    TheFtab.SetValue(TheField,rec,0)
    TheFtab.SetValue(LabelFld,rec,"Node (manhole)")
  else
  end
else
’ Junc = GraphicShape.Make(TheFtab.ReturnValue(ShpFld,rec))
’ NewSym = Junc.GetSymbol
’ NewSym.SetSize(8)
’ NewSym.SetColor(color.getRed)
’ Theview.GetGraphics.Add(Junc)
  TheType = TheFtab.ReturnValue(TypeFld,rec)
  If (TheType = 1) then
    TheFtab.SetValue(TheField,rec,1)
    TheFtab.SetValue(LabelFld,rec,"Junction (no manhole)")
  elseif (TheType = -1) then
    TheFtab.SetValue(TypeFld,rec,5)
    TheFtab.SetValue(LabelFld,rec,"Node (manhole)")
  elseif (TheType = 5) then
    TheFtab.SetValue(TypeFld,rec,-1)
    TheFtab.SetValue(LabelFld,rec,"Node (no manhole)")
  end
end

TheLegend = TheTheme.GetLegend
TheTheme.UpdateLegend

TheFtab.SetEditable(false)

’---------------------------------------------
’---------------END SCRIPT---------------------
’---------------------------------------------
```

```
’----------------------------------------------
’---------Creation Information-----------------
’----------------------------------------------
’
’Name: osortpt
’Author:  Cindy How
’Date:    6/25/97
’
’----------------------------------------------
’------------Purpose/Description---------------
’----------------------------------------------
’
’This script will number the points which
’are not terminal nodes or are terminal nodes
’with direct inflow.  These are the nodes which
’should be numbered under the naUTilus numbering
’system
’
’This script should be run after the scripts
’JUNCTION and INFLNODE have been applied.
’
’FOR USE ON OSBL UNITS
’
’-----------------------------------------------
’-------------------Input-----------------------
’-----------------------------------------------
’
’1.  Node theme - onode.shp
’2.  Attribute table for the theme nodes
’    - field "Node_Type"
’    - field "Node" (empty)
’
’-----------------------------------------------
’------------------Output-----------------------
’-----------------------------------------------
’
’1.  naUTilus node numbers in the "Node" field
’
’-----------------------------------------------
’---------Get Initial Information---------------
’-----------------------------------------------
TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme= TheView.FindTheme("Onode.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.FindField("Onodes")
TypeFld = TheFtab.FindField("Node_Type")
NodeFld = TheFtab.FindField("Node")
TheFtab.SetEditable(True)




k = 1
for each rec in TheFtab
  type = TheFtab.ReturnValue(TypeFld,rec)
  if (type = -1) then
    TheFtab.SetValue(NodeFld,rec,0)
  elseif (type = 2) then
    TheFtab.SetValue(NodeFld,rec,-1)
  else
    TheFtab.SetValue(NodeFld,rec,k)
    k=k+1
  end
end


’----------------------------------------------
’---------------END SCRIPT---------------------
’----------------------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: obrtab
'Author:  Cindy How
'Date:    6/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will use the branch theme of the
'ISBL system and, like DRAININPUT and CONNECTIVITY,
'continue to create the input file by describing
'the reach/branch characteristics as naUTilus requires.
'
'The required data is copied from the branch
'attribute table to a new table called obrtab.dbf
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  Attribute table for the theme obranch
'2.  Attribute table for the theme onode
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
'1.  New table called obrtab.dbf holding the information
'    on branch diameter, length, and slope needed
'    to create part of the naUTilus input file.
'
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheView = av.getActiveDoc
BrTheme = theView.Findtheme("obranch.shp")
BrTab = BrTheme.GetFtab
NdTheme = theView.Findtheme("onode.shp")
NodeTab = NdTheme.GetFtab


'Find Fields:

BFld = BrTab.FindField("Osbl_")
LFld = BrTab.FindField("Length")

TFld = NodeTab.FindField("Node_type")
NFld = NodeTab.FindField("Node")
AFld = NodeTab.FindField("Ex_br")
ArcFld = NodeTab.FindField("Arc_")

NodeTab.SetEditable(true)
'----------------------------------------------
'-------------Add New Table/Fields--------------
'----------------------------------------------


NewName = "obrtab.dbf".AsFilename
NewTab = VTab.MakeNew(NewName,dBase)

BNumFld = Field.Make("Branch",#Field_short,4,0)
NewLFld = Field.Make("Length",#Field_Float,8,4)
DiamFld = Field.Make("Diameter",#Field_Float,8,4)
SlopeFld = Field.Make("Slope",#Field_Float,8,4)
NewTab.AddFields({BNumFld,NewLfld,DiamFld,SlopeFld})
NewTab.SetEditable(true)


'----------------------------------------------
'---------------Prompt for input----------------
'----------------------------------------------


defd = 1.0
defs = 0.01

DDiam = msgbox.input("Please enter the default diameter.","Default
diameter",defd.AsString)
Dslope = msgbox.input("Please enter the default slope.","Default
slope",defS.AsString)

numbr = 0
For each record in BrTab 'fill table with default values
  NewTab.AddRecord
```

```
  brlength = BrTab.ReturnValue(LFld,record)
  brnum = BrTab.ReturnValue(BFld,record)
  NewTab.SetValue(BNumFld,record,brnum)
  NewTab.SetValue(NewLfld,record,brlength)
  NewTab.SetValue(DiamFld,record,DDiam)
  NewTab.SetValue(SlopeFld,record,DSlope)
  numbr = numbr + 1
end


For each record in NodeTab
  type = NodeTab.ReturnValue(TFld,record)
  if (type = 5) then
    NewTab.AddRecord
    brnum = numbr + 1
    NodeTab.SetValue(AFld,record,brnum)
    NewTab.SetValue(BNumFld,numbr,brnum)
    NewTab.SetValue(NewLFld,numbr,1)
    NewTab.SetValue(DiamFld,numbr,defd)
    NewTab.SetValue(SlopeFld,numbr,defs)
    numbr = numbr + 1
  else
    typ = NodeTab.ReturnValue(TFld,record)
    if (typ = -1) then
      Br = NodeTab.ReturnValue(ArcFld,record)
      NodeTab.SetValue(AFld,record,br)
    else
    end
  end
end

NodeTab.SetEditable(False)
NewTab.SetEditable(False)
TheTab = Table.Make(NewTab)
TheTab.SetName("obrtab.dbf")

msgbox.info("Click on the branches you wish to edit.","User Input")
'-----------------------------------------------
'---------------------END-----------------------
'-----------------------------------------------
```

```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'
'Name: oselbr
'Author:  Cindy How
'Date:    6/97
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take select the branch to be
'editted.  It is to be used as the apply event
'with OBRTAB.
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  "obrtab.dbf" file
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
'1.  Selected record
'
'----------------------------------------------
'---------Get Initial Information---------------
'----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme = TheView.FindTheme("obranch.shp")
TheTabName = av.GetProject.FindDoc("obrtab.dbf")
TheTab = TheTabName.GetVtab
DiamFld = Thetab.Findfield("Diameter")
SlopeFld = Thetab.FindField("Slope")
TheTab.SetEditable(true)

NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Thetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
```

```
    Msgbox.error("That point is not found, try again","Not found")
    exit
  end
rec=PtOnTheme.get(0)


OldDia = TheTab.ReturnValue(DiamFld,rec)
OldSlope = TheTab.ReturnValue(SlopeFld,rec)
labels = {"Diameter (m)","Slope"}
defaults = {OldDia.AsString,OldSlope.AsString}
NewValList = msgbox.multiInput("Enter the branch characteristics","New
Values",labels,defaults)
test = NewValList.Count
If (test > 0) then
  NewDiam = NewValList.get(0).AsNumber
  NewSlope = NewValList.Get(1).AsNumber
  TheTab.SetValue(DiamFld,rec,NewDiam)
  TheTab.SetValue(SlopeFld,rec,NewSlope)
else
end

TheTab.SetEditable(false)

’-----------------------------------------------
’--------------------END----------------------
’-----------------------------------------------
```



```
’---------------------------------------------
’---------Creation Information-----------------
’---------------------------------------------
’Name: odroptab
’Author:  Cindy How
’Date:    7/2/97
’
’---------------------------------------------
’-----------Purpose/Description---------------
’---------------------------------------------
’
’This program will create a table called drops.dbf.
’It is to be used as the click event with the
’script ODROPDAT as a apply event.
’
’---------------------------------------------
’------------------Input----------------------
’---------------------------------------------
’
’1.  Theme onode.shp
’2.  Attribute table of onode.shp
’3.  Theme obranch.shp
’4.  Attribute table of obranch.shp
’
’---------------------------------------------
’-----------------Output----------------------
’---------------------------------------------
’
’1.  Empty table called odrops.dbf
’
’---------------------------------------------
’---------Getting initial data------------------
’---------------------------------------------
theProject=av.getProject
TheTab = TheProject.FindDoc("odrops.dbf")
flag = 0
if (TheTab.AsString = "nil") then
  TabName="odrops.dbf".AsFilename
  NewTable=VTab.MakeNew(TabName,dBase)
  flag = 1
else
  NewTable = TheTab.GetVtab
end
NewTable.SetEditable(True)
```

```
'------------------------------------------------
'-------Adding Fields to the New Table----------
'------------------------------------------------

if (flag = 1) then
  NFld=Field.make("Node",#Field_long,8,0)
  BFld=Field.make("Branch",#Field_long,8,0)
  HFld=Field.make("Height",#Field_decimal,10,4)
  TFld=Field.make("Tailwater",#Field_decimal,10,4)
  NewTable.AddFields({NFld,BFld,HFld,TFld})
else
check = msgbox.longyesno("Reset drops?  (Delete existing drops)","User
Input",true)
  if (check = true) then
    for each record in NewTable
      NewTable.RemoveRecord(record)
    end
  else
  end
end

'Add Table
TheTab = Table.Make(NewTable)
TheTab.SetName("odrops.dbf")

NewTable.SetEditable(false)

MsgBox.Info("Click on each branch where a drop occurs. When done, move
on.","User Input")


'------------------------------------------------
'---------------------END----------------------
'------------------------------------------------
```



```
'-----------------------------------------------
'---------Creation Information-----------------
'-----------------------------------------------
'
'Name: odropdat
'Author:  Cindy How
'Date:    7/2/97
'
'-----------------------------------------------
'------------Purpose/Description---------------
'-----------------------------------------------
'
'This program was created from the ISBL script
'dropdata.  It is to be run in conjunction with
'the script ODROPTAB.  It will add a record to
'the odrop.dbf table for each drop and
'fill in values of that table.  The user will be
'prompted to select or accept some default values.
'
'------------------------------------------------
'------------------Input----------------------
'------------------------------------------------
'
'1.  Empty table odrops.dbf
'2.  Theme onode.shp
'3.  Attribute table of onode.shp
'4.  Theme obranch.shp
'5.  Attribute table of obranch.shp
'
'------------------------------------------------
'-----------------Output----------------------
'------------------------------------------------
'
'1.  Filled table called odrops.dbf
'
'------------------------------------------------
'---------Getting initial data------------------
'------------------------------------------------
theProject=av.getProject
TheView = av.GetActiveDoc
BrTheme = theView.FindTheme("obranch.shp")
BrTab = BrTheme.GetFtab
NdTheme = TheView.FindTheme("onode.shp")
```

```
NdTab = NdTheme.GetFtab
TabName=TheProject.FindDoc("odrops.dbf")
if (Tabname.asString = "nil") then
  TabName = "odrops.dbf".asFilename
else
  TheProject.RemoveDoc(TabName)
end

DropTab = TabName.GetVtab
DropTab.Seteditable(true)
'
'-----------------------------------------------
'----------------Find Fields--------------------
'-----------------------------------------------

'InDropTab
NFld = DropTab.FindField("Node")
BFld = DropTab.FindField("Branch")
HFld = DropTab.FindField("Height")
TFld = DropTab.FindField("Tailwater")

'InNdTab
NdFld = NdTab.FindField("Node")

'InBrTab
BrFld = BrTab.FindField("osbl_")
TNdFld = BrTab.FindField("Tnode_")

'------------------------------------------------
'-------------Set Default Values-----------------
'------------------------------------------------
height = 1
depth = 0.5


'------------------------------------------------
'---------------Get Branch Data------------------
'------------------------------------------------


NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = BrTheme.FindByPoint(NewPt)
If (PtOnTheme.IsEmpty) then
  MsgBox.Error("That point is not found, try again","Not found")
  DropTab.SetEditable(False)
  TheTab = Table.Make(DropTab)
  TheTab.SetName("odrops.dbf")
  exit
end
rec = PtOnTheme.Get(0)

BrNum = BrTab.ReturnValue(BrFld,rec)
Tnode = BrTab.ReturnValue(TNdFld,rec)
NodeRec = Tnode - 1


'-----------------------------------------------
'---------------Get Node Data-------------------
'-----------------------------------------------

NNodeNum = NdTab.ReturnValue(NdFld,NodeRec)

'----------------------------------------------
'------------Add and fill records---------------
'----------------------------------------------

BrList = List.Make

count = 0
For each record in DropTab
  count = count + 1
  TheBranch = DropTab.ReturnValue(BFld,record)
  BrList.Add(TheBranch)
end
'msgbox.listAsString(BrList,"branches with drops","test")

check2 = BrList.FindByValue(BrNum)
if (check2 > -1) then
  change = msgbox.longyesno("A drop already exists for that branch.  Edit
chracteristics?","User Input",true)
  if (change.asString = "true") then
    labels = {"Height (m)","Tailwater depth (m)"}
    height = DropTab.ReturnValue(HFld,check2)
    depth = DropTab.ReturnValue(TFld,check2)
    defaults = {height.AsString,depth.AsString}
    OutList = MsgBox.MultiInput("Enter the drop characteristics","User
Input",labels,defaults)
    test = outlist.count
```

```
    if (test > 0) then
      Ht = OutList.Get(0).AsNumber
      Twd = Outlist.Get(1).AsNumber
    else
      msgbox.info("No changes made.","No changes")
      DropTab.SetEditable(False)
      TheTab = Table.Make(DropTab)
      TheTab.SetName("odrops.dbf")
      exit
    end
    DropTab.SetValue(HFld,check2,ht)
    DropTab.SetValue(TFld,check2,twd)
  else
    msgbox.info("No changes made.","No changes")
    DropTab.SetEditable(False)
    TheTab = Table.Make(DropTab)
    TheTab.SetName("odrops.dbf")
    exit
  end
else


  labels = {"Height (m)","Tailwater depth (m)"}
  defaults = {height.AsString,depth.AsString}
  OutList = MsgBox.MultiInput("Enter the drop characteristics","User
Input",labels,defaults)

  test = OutList.Count

  If (test > 0) then
    DropTab.AddRecord
    Ht = OutList.Get(0).AsNumber
    Twd = Outlist.Get(1).AsNumber
    if (count > 0) then
        recnum = count
      else
        recnum = 0
    end
    DropTab.SetValue(NFld,recnum,NNodeNum)
    DropTab.SetValue(BFld,recnum,BrNum)
    DropTab.SetValue(HFld,recnum,Ht)
    DropTab.SetValue(TFld,recnum,Twd)
  Else
  MsgBox.Info("No drop added at that location.","No Drop")
  end

  TheTab = Table.Make(DropTab)
  TheTab.SetName("odrops.dbf")

  DropTab.SetEditable(false)
end



'----------------------------------------------
'---------------------END----------------------
'----------------------------------------------
```

```
'---------------------------------------
'---------Creation Information-----------
'---------------------------------------
'
'Name: oinflow
'Author:  Cindy How
'Date:    6/27/97
'Revisions:
'1.  8/11/97 by Cindy How
'    - Editted to look for naUTilus output
'      of ISBL units in each ISBL directory
'
'---------------------------------------
'-----------Purpose/Description----------
'---------------------------------------
'
'This script will give the user two options
'on entering data for inflow to the OSBL
'system:  1) User specified input at locations
'clicked on or 2) Information taken from ISBL
'subdirectories with possible additional inflow.
'(for the present, as the structure for option
'2 is not yet ready, option one will be programmed)
'This is to be used with OSBL units.
'
'This script will run as the Click event
'with the script OSELFLOW.
'
'---------------------------------------
'-----------------Input------------------
'---------------------------------------
'
'1.  Node theme Onodes.shp
'2.  Attribute table to node theme
'    - Field "Ex_br"
'
'---------------------------------------
'-----------------Output-----------------
'---------------------------------------
'
'1.  oinflow.dbf file with branch, flow,
'    temperature, concentration, and oil
'    fraction for each inflow site (empty
'    fields).
'
'This file will help create the OSBL.IN
'naUTilus input file
'
'---------------------------------------
'-----------Get Initial Info-------------
'---------------------------------------
'
TheProject = av.GetProject
TheView = av.GetActiveDoc
TheISBLlist = TheProject.FindDoc("isbllist.dbf")
TheListWin = TheISBLlist.GetWin
TheListWin.Open
TheISBLTab = TheISBLList.GetVtab
BrFld = TheISBLTab.FindField("Branch")
NameFld = TheISBLTab.FindField("ISBL_Name")
FRFld = TheIsblTab.FindField("Flow_rate")
LCFld = TheISBLTab.FindField("Liq_conc")
TempFld = TheISBLTab.FindField("Temp")
OFFld = TheISBLTab.FindField("Oil_frac")
ASEFld = TheISBLTab.FindField("Ab_Sewer_em")
TMEFld = TheISBLTab.FindField("Tot_mass_in")
EMFld = TheISBLTab.FindField("Emission_rate")
FldLst=List.Make
if (ASEFld.AsString = "nil") then
  ASEFld = Field.Make("Ab_Sewer_em",#Field_decimal,10,5)
  FldLst.Add(ASEFld)
else
end
if (TMEFld.AsString = "nil") then
  TMEFld = Field.Make("Tot_mass_in",#Field_decimal,10,5)
  FldLst.Add(TMEFld)
else
end
if (EMFld.AsString = "nil") then
  EMFld = Field.Make("Emission_rate",#Field_decimal,11,6)
  FldLst.Add(EMFLd)
else
end
TheISBLTab.SetEditable(true)
testcount = FldLst.Count
if (testcount > 0) then
  TheISBLTab.AddFields(FldLst)
```

```
else
end

ResetChoice = MsgBox.LongYesNo("Reset any existing inflow data?","User
Input",true)
if (resetChoice.AsString = "true") then
  for each record in TheISBLTab
    TheISBLTab.SetValue(BrFld,record,0)
  end
else
end

TheMainDir = FN.GetCWD

’check for table already exists
TheTable = TheProject.FindDoc("oinflow.dbf")
if (TheTable.asString = "nil") then
  Flag = msgbox.longYesNo("Are there inflows other than the listed ISBL
inflows?","User Input",true)
  if (flag.AsString = "true") then

    ’Add New Table
    Tabname = "oinflow.dbf".AsFilename
    NewTable = VTab.MakeNew(Tabname,dBase)

    ’----------------------------------------
    ’-----Adding Fields to the New Table-----
    ’----------------------------------------

    BFld = Field.Make("Branch",#Field_short,6,0)
    FFld = Field.Make("Flow",#Field_decimal,10,4)
    TFld = Field.Make("Temperat",#Field_decimal,8,3)
    CFld = Field.Make("Concentra",#Field_decimal,10,4)
    OFld = Field.Make("Oil_frac",#Field_decimal,7,5)
     NewTable.AddFields({BFld,FFld,TFld,CFld,OFld})

    ’AddTable
    TheTable = Table.Make(NewTable)
    TheTable.SetName("oinflow.dbf")
    NewWin = TheTable.GetWin
    NewWin.Open
  else
  end
else
  NewWin = TheTable.getWin
  NewWin.Open

end

For each record in TheISBLtab
  DirName = TheISBLTab.ReturnValue(NameFld,record).AsFilename
  DirName.SetCWD
  DocName = "isblout.txt".AsFilename
  NewTab = Vtab.Make(DocName,false,false)
  Fld1 = NewTab.FindField("Flow_rate")
  Fld2 = NewTab.FindField("Liq_conc")
  Fld3 = NewTab.FindField("Temp")
  Fld4 = NewTab.FindField("Oil_frac")
  Fld5 = NewTab.FindField("Above_sew_emm")
  Fld6 = NewTab.FindField("Tot_mass_in")
  Fld7 = NewTab.FindField("Emission_rate")
  val1 = NewTab.ReturnValue(Fld1,0)
  val2 = NewTab.ReturnValue(Fld2,0)
  val3 = NewTab.ReturnValue(Fld3,0)
  val4 = NewTab.ReturnValue(Fld4,0)
  val5 = NewTab.ReturnValue(Fld5,0)
  val6 = NewTab.ReturnValue(Fld6,0)
  val7 = NewTab.ReturnValue(Fld7,0)
  TheISBLTab.SetValue(FRFld,record,val1)
  TheISBLTab.SetValue(LCFld,record,val2)
  TheISBLTab.SetValue(TempFld,record,val3)
  TheISBLTab.SetValue(OFFld,record,val4)
  TheISBLTab.SetValue(ASEFld,record,val5)
  TheISBLTab.SetValue(TMEFld,record,val6)
  TheISBLTab.SetValue(EMFld,record,val7)
  TheMainDir.SetCWD
end
TheISBLTab.SetEditable(false)

MsgBox.Info("Click on a point where flow occurs (ISBL or other)","User Input")
ViewWin = TheView.GetWin
ViewWin.Activate

’-----------------------------------------------
’---------------------END-----------------------
’-----------------------------------------------
```

```
’---------------------------------------
’---------Creation Information-----------
’---------------------------------------
’
’Name: oflowsel
’Author:  Cindy How
’Date:    6/27/97
’
’---------------------------------------
’-----------Purpose/Description----------
’---------------------------------------
’
’This script will allow the user to enter
’data for inflow to the OSBL system,
’specifying input at locations the user
’clicks on.
’
’This script runs as the Apply event
’with the script OINFLOW.
’
’NOTE:  This must be changed to deal with
’the user selecting an "internal node" (not
’a "terminal" node.
’
’---------------------------------------
’-----------------Input------------------
’---------------------------------------
’
’1.  Node theme Onodes.shp
’2.  Attribute table to node theme
’    - Field "Ex_br"
’3.  Table oinflow.dbf
’
’---------------------------------------
’-----------------Output-----------------
’---------------------------------------
’
’1.  oinflow.dbf file with branch, flow,
’    temperature, concentration, and oil
’    fraction for each inflow site.
’
’This file will help create the OSBL.IN
’naUTilus input file
’
’---------------------------------------
’-----------Get Initial Info-------------
’---------------------------------------
’
TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme = TheView.FindTheme("onode.shp")
TheFtab = TheTheme.GetFtab
TheTabName = theProject.FindDoc("oinflow.dbf")

If(TheTabName.AsString = "nil") then

else
  FlowTab = TheTabName.GetVtab

  FlowTab.SetEditable(True)

  ’in FlowTab
  BFld = FlowTab.FindField("Branch")
  FFld = Flowtab.FindField("Flow")
  TFld = Flowtab.FindField("Temperat")
  CFld = Flowtab.FindField("Concentra")
  OFld = Flowtab.FindField("Oil_frac")
end

’FindFields
’in theFtab
BrFld = TheFtab.FindField("Ex_br")

ISBLTab = TheProject.FindDoc("isbllist.dbf").GetVtab
NameFld = ISBLTab.FindField("ISBL_Name")
ConBrFld = ISBLTab.FindField("Branch")

TheODB = ODB.Open("permdat.odb".AsFilename)
ISBLlist = TheODB.Get(0)

’---------------------------------------
’----------User Select Point-------------
’---------------------------------------

NewPt = TheView.GetDisplay.ReturnUserPoint
PtOnTheme = Thetheme.FindByPoint(NewPt)
  If (PtOnTheme.isEmpty) then
```

```
    Msgbox.error("That point is not found, try again","Not found")
    exit
  end
rec=PtOnTheme.get(0)

branch = TheFtab.ReturnValue(BrFld,rec)

if (branch = 0) then 'internal node
  msgbox.info("That branch is an internal node.  Not a valid entry","Not Valid")
  exit
else
end

ct = 0
BrList = List.Make

If (TheTabName.AsString = "nil") then
else
  For each record in FlowTab
    ct = ct + 1
    br = FlowTab.ReturnValue(BFld,record)
    BrList.Add(br)
  end
end

For each record in ISBLTab
  ISBLnd = ISBLTab.ReturnValue(ConBrFld,record)
  if (ISBLnd > 0) then
    BrList.Add(ISBLnd)
  else
  end
end

Idx = BrList.FindByValue(branch)


If (idx = -1) then

  ChoiceList = {"ISBL","Other Inflow"}
  opt = msgbox.choiceAsString(ChoiceList,"Please indicate what is located at this
node.","User Input")
  ChoiceIdx = ChoiceList.FindByValue(opt)

  If (choiceIdx = 1) then 'Other Inflow
    if (thetabname.AsString = "nil") then
      msgbox.info("No non-ISBL flow indicated, no flow added.","Not Valid")
      exit
    else
    end
    labels = {"Flow (L/s)","Temperature (C)","Concentration (mg/L)","Oil
fraction"}
    defaults = {"1.0","20","5","0.0"}
    OutList = msgBox.MultiInput("Enter inflow characteristics","User
Input",labels,defaults)

    Test = Outlist.Count

    If (Test > 0) then
      Flow = Outlist.Get(0).AsNumber
      Temp = Outlist.Get(1).AsNumber
      Conc = Outlist.Get(2).AsNumber
      Frac = Outlist.Get(3).AsNumber
      FlowTab.AddRecord
      recnum = ct
      FlowTab.SetValue(BFld,recnum,branch)
      FlowTab.SetValue(FFld,recnum,flow)
      FlowTab.SetValue(TFld,recnum,temp)
      FlowTab.SetValue(CFld,recnum,conc)
      FlowTab.SetValue(OFld,recnum,frac)
      FlowTab.SetEditable(False)
    else
      MsgBox.Info("No inflow information added to table","No Input")
    end
  elseif (ChoiceIdx = 0) then 'ISBL location

    TheISBL = Msgbox.ChoiceAsString(ISBLlist,"Which unit is at the selected
point?","User Input")
    ISBLidx = ISBLlist.FindByValue(TheISBL)
    ISBLTab.SetEditable(true)
    checklist = list.Make
    For each rec in ISBLTab
      TheBr = ISBLTab.ReturnValue(ConBrFld,rec)
      Checklist.Add(TheBr)
    end
    brfind = CheckList.FindByValue(Branch)


    'TAG - FIX THIS SECTION!!
```

```
    if (brfind = -1) then
      ISBLTab.SetValue(ConBrFld,ISBLidx,branch)
    else
      oldISBL = ISBLTab.ReturnValue(NameFld,brfind)
      optionlist = {"Change ISBL","Keep present ISBL"}
      Opt = MsgBox.ChoiceAsString(optionlist,"The ISBL"++OldISBL++"is
located there.  Update?","User Input")
      OptIdx = OptionList.FindByValue(opt)
      if (optIdx = 0) then
        ISBLTab.SetValue(ConBrFld,ISBLidx,branch)
        ISBLTab.SetValue(ConBrFld,brfind,0)
      else
        msgbox.info("ISBL location not set/changed","No Change.")
      end
    end


    ISBLTab.SetEditable(false)
  else
    msgbox.info("No flow added.","User Cancel")
  end
else
  MsgBox.Info("Flow already exists at that location.  No flow added","No Input")
end


'-----------------------------------------------
'---------------------END-----------------------
'-----------------------------------------------
```



```
'---------------------------------------------
'---------Creation Information-----------------
'---------------------------------------------
'

'Name: osblinp
'Author:  Cindy How
'Date:    6/97
'Revisions:
'1.  8/11/97 by Cindy How
'    - Editted to read inflow data from
'      both sources (ISBLLIST.DBF, OINFLOW.DBF).
'
'---------------------------------------------
'------------Purpose/Description---------------
'---------------------------------------------
'
'This script will take the .dbf files created
'in previous steps and create the naUTilus input
'file.  The input file will be called "osbl.in"
'
'----------------------------------------------
'-------------------Input-----------------------
'----------------------------------------------
'
'1.  Node theme
'2.  Attribute table of the Node theme
'3.  Branch theme
'4.  Attribute table of the Branch theme
'5.  oinflow.dbf table
'6.  osblcon.dbf table
'7.  obr.dbf table
'8.  odrops.dbf table
'9.  isbllist.dbf table
'
'----------------------------------------------
'------------------Output-----------------------
'----------------------------------------------
'
'1.  Text input file, "osbl.in"
'
'
'----------------------------------------------
'-------------------Notes-----------------------
'----------------------------------------------
```

```
'
'The variables InList, InDefault, and OutList are
'temporary and used at several points in the script.
'They take on new values several times.
'
'-----------------------------------------------
'---------Get Initial Information---------------
'-----------------------------------------------

TheProject = av.GetProject
TheView = av.GetActiveDoc
NdTheme = theview.Findtheme("onode.shp")
BrTheme = TheView.Findtheme("obranch.shp")
NodeTab = NdTheme.GetFtab
BranchTab = BrTheme.GetFtab
FTabName = TheProject.FindDoc("oinflow.dbf")

av.run("osblconn",{})

If (FTabName.AsString = "Nil") then
  flag = 1
else
  flag = 0
  FlowTab = FTabName.GetVtab
end
CTabName = TheProject.FindDoc("osblcon.dbf")
ConnTab = CTabName.GetVtab
BTabName = TheProject.Finddoc("obrtab.dbf")
BrTab = BTabName.GetVtab
DropName = TheProject.FindDoc("odrops.dbf")
DropTab = DropName.GetVtab
ISBLname = TheProject.FindDoc("isbllist.dbf")
ISBLTab = ISBLname.GetVtab

'-----------------------------------------------
'------------------Find Fields------------------
'-----------------------------------------------
'In BrTab
BFld = BrTab.FindField("Branch")
LFld = BrTab.Findfield("Length")
DFld = BrTab.Findfield("Diameter")
SFld = BrTab.FindField("Slope")

'In flowtab
If (flag = 0) then
  ConFld = flowtab.FindField("Branch")
  FlFld = flowtab.FindField("Flow")
  TempFld = flowtab.Findfield("Temperat")
  CFld = flowtab.FindField("Concentra")
  OilFld = flowtab.FindField("Oil_frac")
Else
end

'InConnTab
Fld1 = ConnTab.FindField("Cola")
Fld2 = ConnTab.FindField("Colb")

'InNodeTab
NNumFld = NodeTab.Findfield("Node")
ArcFld = NodeTab.Findfield("Arc_")
TypeFld = NodeTab.Findfield("Node_Type")
JFld = NodeTab.FindField("Junction")

'InBranchTab
BrFld = BranchTab.FindField("Osbl_")
FFld = BranchTab.Findfield("Fnode_")

'InDropTab
DNdFld = DropTab.FindField("Node")
DBrFld = DropTab.Findfield("Branch")
DHtFld = DropTab.FindField("Height")
DTWDFld = DropTab.FindField("Tailwater")

'InISBLTab
IConFld = ISBLTab.FindField("Branch")
IFlFld = ISBLTab.FindField("Flow_rate")
ITempFld = ISBLTab.FindField("Temp")
ICFld = ISBLTab.FindField("Liq_conc")
IOilFld = ISBLTab.FindField("Oil_frac")

'-----------------------------------------------
'----------------Make Text File-----------------
'-----------------------------------------------

NewFile = "osbl.in".AsFileName
LFile = LineFile.Make(NewFile,#File_Perm_write)

Lfile.WriteElt("Number of branches")
```

```
numbr = 0
For each record in Brtab
  numbr = numbr + 1
end
LFile.WriteElt(numbr.AsString)

LFile.WriteElt("Number of Nodes")
numnode = 0
juncnt = 0
For each record in Nodetab
  Nd = NodeTab.ReturnValue(NNumFld,record)
  if (nd < 1) then
  else
    numnode = numnode + 1
  end
  Junc = NodeTab.ReturnValue(JFld,record)
  if (Junc = 1) then
    juncnt = juncnt + 1
  else
  end
end
LFile.WriteElt(numnode.AsString)

Lfile.WriteElt("Last branch, Last node")

'finds the last node in the system by the naUTilus numbering system
arclist = list.make
TypeList = list.make
For each record in NodeTab
  arcnum = NodeTab.ReturnValue(ArcFld,record)
  ntype = NodeTab.ReturnValue(TypeFld,record)
  ArcList.Add(arcnum)
  TypeList.Add(ntype)
end
IdxNum = TypeList.FindByValue(2)
arcid = ArcList.Get(IdxNum)
arcidx = arcid - 1
LNodeAv = BranchTab.ReturnValue(FFld,arcidx)
LNodeAvIdx = LNodeAv - 1
LastNode = NodeTab.ReturnValue(NNumFld,LNodeAvIdx)

'finds last branch in the system.
LastBranch = NodeTab.ReturnValue(ArcFld,IdxNum)

LFile.WriteElt(LastBranch.AsString+","+LastNode.AsString)

'TAG! Need to count number of manholes
numman = numnode - juncnt
LFile.WriteElt("Number of manhole covers, number of openings per manhole,
opening area (m2)")

'User prompt for manhole characteristics
InList = {"No. of openings per cover","opening area (m2)"}
InDefault = {"4","5.1e-4"}
OutList = MsgBox.MultiInput("Enter manhole characteristics","User
Input",InList,InDefault)
listcount = Outlist.Count
If (listcount = 0) then
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  Exit
else
  'write manhole characteristics

LFile.WriteElt(numman.AsString+","+Outlist.Get(0).AsString+","+Outlist.Get(1)
.AsString)
end

'TAG-Need to count number of junctions with no manholes, enter node numbers
LFile.WriteElt("Number of junctions with no manholes")
LFile.WriteElt(juncnt.AsString)
LFile.WriteElt("Enter node number for junctions with no manholes")

For each record in NodeTab
  junc = NodeTab.ReturnValue(JFld,record)
  If (junc = 1) then
    JuncNode = NodeTab.ReturnValue(NNumFld,record)
    LFile.WriteElt(JuncNode.AsString)
  else
  end
end

'User Prompt for wind speed
LFile.WriteElt("Ambient wind velocity (m/s)")
wspeed = MsgBox.Input("Enter the ambient wind speed (m/s)","User Input","1")
if (wspeed.asString = "nil") then
```

```
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  exit
else
  'write wind speed
  LFile.WriteElt(wspeed.AsString)
end

'User prompt for ambient conditions
InList = {"Ambient Temperature (C)","Relative humidity"}
InDefault = {"25","0.5"}
OutList = MsgBox.MultiInput("Enter the Ambient Conditions","User
Input",InList,InDefault)
listcount = Outlist.Count
If (listcount = 0) then
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  Exit
else
  'write ambient conditions
  LFile.WriteElt("Ambient Temperature (C) and relative humidity")
  LFile.WriteElt(OutList.Get(0).AsString+","+OutList.Get(1).AsString)
end


'prompt user to input Henry's law method

MethodList = {"Hc=exp(A-B/T)","Hc at known T, adjusted using Antoine's
constants","VP/Sol using Antoine's constants","Hc=constant"}
'MethodList = {"1","2","3","4"}
listcount = Outlist.Count
If (listcount = 0) then
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  Exit
else
  meth = Msgbox.ChoiceAsString(MethodList,"Select the method used to
determine Henry's constant:","Henry's law Constant")
  MethodIdx = MethodList.FindByValue(meth)
  Method = MethodIdx + 1

  LFile.WriteElt("Henry's law Constant - Method 1,2,3,4")
  LFile.WriteElt(Method.AsString)
end
```

```
'Prompt for correct variables (dependent on Method)
If (Method = 1) then
  InLabels = {"A","B"}
  InDefaults = {"5.53","3194"}
  OutList = Msgbox.MultiInput("Enter the values of A and B (Hc = exp(A-
B/T))","Method 1",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then
    LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
  exit
  end
elseif (Method = 2) then
  InLabels = {"Hc","T in C","Antoine's A","Antoine's B","Antoine's C"}
  InDefaults = {"0.5","25","1","1","1"}
  Outlist = Msgbox.MultiInput("Enter the Values of Hc at a known temperature,
the known temperature, and Antoine constants A,B, and C.  (Antoine constants for
T in K, giving vapor Presure in mmHg","Method 2",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString+","+Outlist.Get(3).AsString+","+Outlist.Get(4).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
  exit
  end
elseif (Method = 3) then
  InLabels = {"A","B","C","Solubility (g/L)","Molecular Weight (g/mol)"}
  InDefaults = {"1","1","1","1","100"}
  OutList = MsgBox.MultiInput("Enter the values of Antoine's A, B, and C,
solubility, and MW","Method 3",InLabels,InDefaults)
  check = OutList.Count
  If (check > 0) then

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString+","+Outlist.Get(3).AsString+","+Outlist.Get(4).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
```

```
  exit
  end
elseif (Method = 4) then
  InLabels = {"Hc"}
  InDefaults = {"0.5"}
  OutList = MsgBox.MultiInput("Enter the Henry’s Constant","Method
4",Inlabels,InDefaults)
  check = OutList.Count
  If (check > 0) then
    LFile.WriteElt(Outlist.Get(0).AsString)
  else
    MsgBox.Info("Error: Input file will not be complete.  Re-run the
script.","Error")
  exit
  end
else
  MsgBox.Info("Error: Input file will not be complete.  Re-run the script.","Error")
  exit
end

’User Prompt for diffusivities
InList = {"Liquid Phase Diffusivity (cm2/s)","Gas Phase Diffusivity (cm2/s)"}
InDefault = {"1.0e-5","0.1"}
outList = MsgBox.MultiInput("Enter the Diffusivities","User
Input",InList,InDefault)
listcount = Outlist.Count
If (listcount = 0) then
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  Exit
else
  ’write diffusivities
  Lfile.WriteElt("Liquid Diffusivity, Gas Diffusivity")
  LFile.WriteElt(OutList.Get(0).AsString+","+OutList.Get(1).AsString)
end


’User prompt for presence of oil
oilList = {"No oil present","Oil present only at surface","Oil present and dispersed
(not at surface)"}
oilcheck = MsgBox.ChoiceAsString(OilList,"Indicate presence and condition of
oil","User Input").AsString
If (oilcheck.asString = "nil") then
  Msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  exit
else

  oilIdx = OilList.FindbyValue(oilcheck)
  oilcond = oilidx + 1
  LFile.WriteElt("1 = no oil, 2 = oil only at surface")
  LFile.WriteElt("3 = oil only dispersed, not at surface")
  LFile.WriteElt(oilcond.AsString)
  LFile.WriteElt("If oil is present, read Kow, density, oil MW (else, empty)")
end

if (oilcond > 1) then
  InLabels = {"Kow","Oil density (g/m3)","Oil molecular weight (g/mol)"}
  InDefaults = {".5","20","100"}
  OutList = MsgBox.MultiInput("Enter oil characteristics","User
Input",InLabels,InDefaults)
  listcount = Outlist.Count
  If (listcount = 0) then
    msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
    Exit
  else

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString)
  end
else
end


’fill dummy variable lines (5 of them)
LFile.WriteElt("Indicate method of determining vapor pressure if oil is present")
LFile.WriteElt("Method 1-Antoine’s constants")
LFile.WriteElt("Enter A,B,C for T in K giving VP in mmHg")
LFile.WriteElt("Method 2-Vapor Pressure = constant")
LFile.WriteElt("Enter Vapor Pressure")

if (oilcond > 1) then
  MethList = {"Antoine’s constants","Constant vapor pressure"}
  oilmeth = msgbox.ChoiceAsString(MethList,"Indicate method of calculating
vapor pressure","User Input").AsString
  methind = MethList.findByValue(oilMeth)
```

```
  indnum = methind + 1
  if (indnum = 1) then
    InLabels = {"A","B","C"}
    InDefaults = {"1","1","1"}
    Outlist = MsgBox.MultiInput("Enter the Antoine's constants (for T in K, giving
vapor pressure in mmHg)","User Input",InLabels,InDefaults)
    listcount = Outlist.Count
    If (listcount = 0) then
      msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
      Exit
    else

LFile.WriteElt(Outlist.Get(0).AsString+","+Outlist.Get(1).AsString+","+Outlist.G
et(2).AsString)
    end
  else
    VapPres = MsgBox.Input("Enter the vapor pressure (atm)","User Input","1")
    If (vappres.AsString = "nil") then
      msgbox.info("Output file will not be complete.  Please run script
again.","Error -Incomplete")
      exit
    else
      Lfile.WriteElt(VapPres.AsString)
    end

  end
else
end


'do not use default reach length, enter length, diameter, and slope for each reach
'may need to change this later (6/11/97)
LFile.WriteElt("Enter 1 to change default reach characteristics")
LFile.WriteElt("1")
LFile.WriteElt("Number of branches to be modified")
Lfile.Writeelt("Number, Diameter, Length, slope for each branch")
LFile.WriteElt(numbr.AsString)

for each record in BrTab
  Br = BrTab.ReturnValue(BFld,record)
  Len = BrTab.ReturnValue(LFld,record)
  Diam = BrTab.ReturnValue(DFld,record)
  Slope = BrTab.ReturnValue(SFld,record)

LFile.WriteELT(br.AsString+","+diam.AsString+","+len.AsString+","+Slope.As
String)
end

'user input - indicate P&P or OEG
LFile.WriteElt("1 = Parkhurst&Pomeroy, 2 = Owens-Edwards-Gibbs")
ChList = {"Parkhurst & Pomeroy","Owens-Edwards-Gibbs"}
ch = Msgbox.ChoiceAsString(ChList,"Calculate using:","User Input").AsString
if (ch = "nil") then
  msgbox.info("Input file will not be complete.  Please run the script
again.","Error - Incomplete")
  exit
else
  ChInd = ChList.FindByValue(ch)
  Chnum = chInd + 1
  LFile.WriteElt(Chnum.AsString)
end

'Fill dummy variables (4 lines)
LFile.WriteElt("Enter no. of branches connected to node i")
LFile.WriteElt("Branch number, kind of connection")
LFile.WriteElt("Enter no. of branches connected to node i + 1, etc...")
LFile.WriteElt("Branch number, kind of connection")
LFile.WriteElt("...1 for inflow, -1 for outflow")

for each record in ConnTab
  columna = ConnTab.ReturnValue(Fld1,record)
  columnb = ConnTab.ReturnValue(Fld2,record)
  If (columnb = 0) then
    Lfile.WriteElt(columna.AsString)
  else
    LFile.WriteElt(columna.AsString+","+columnb.AsString)
  end
end

numdrops = 0
For each record in Droptab
  numdrops = numdrops + 1
end

chList = {"Nakasone","WATER8"}
dmethod = MsgBox.ChoiceAsString(chList,"Choose the method for calculating
emissions at drop structures.","User Input")
```

```
if (dmethod = "nil") then
  msgbox.info("Input file will not be complete.  Please run script again.","Error -
Incomplete")
  exit
else
  ChIndex = ChList.FindByValue(dmethod)
  ChNum = ChIndex + 1
end

LFile.WriteElt("Number of drops, Nakasone(1)/WATER8(2)")
LFile.writeElt("enter node, branch, drop height (m), tailwater depth (m) for each
drop")
LFile.WriteElt(numdrops.AsString+","+ChNum.AsString)

If (numdrops = 0) then
else
  For each record in DropTab
    DrNode = DropTab.ReturnValue(DNdFld,record)
    DrBranch = DropTab.ReturnValue(DBrFld,record)
    DrHeight = DropTab.ReturnValue(DHtFld,record)
    DrTWD = DropTab.ReturnValue(DTWDFld,record)

LFile.WriteElt(DrNode.AsString+","+DrBranch.AsString+","+DrHeight.AsString
+","+DrTWD.AsString)
  end
end

LFile.WriteElt("Number of nodes at which flow enters the system")
FlowCount = 0
If (flag = 0) then
  For each record in FlowTab
    FlowCount = FlowCount + 1
  end
else
end

For each record in ISBLTab
  FlowCount = FlowCount + 1
end

LFile.WriteElt(FlowCount.AsString)
LFile.WriteElt("Enter Branch Number,")
LFile.WriteElt("Flow rate (L/s), Temperature (C), Concentration (mg/L), and oil
fraction by volume")

If (flag = 0) then
  for each record in flowtab
    conn = flowtab.ReturnValue(ConFld,record).AsString
    rate = flowtab.ReturnValue(FlFld,record).AsString
    temp = flowtab.ReturnValue(TempFld,record).AsString
    Conc = flowtab.ReturnValue(CFld,record).AsString
    Oil = flowtab.ReturnValue(OilFld,record).AsString
    LFile.WriteElt(conn+","+rate+","+temp+","+conc+","+oil)
  end
else
end

For each record in ISBLTab
  conn = ISBLTab.ReturnValue(IconFld,record).AsString
  rate = ISBLTab.ReturnValue(IFlFld,record).AsString
  Temp = ISBLTab.ReturnValue(ITempFld,Record).AsString
  Conc = ISBLTab.ReturnValue(ICFld,record).AsString
  Oil = ISBLTab.ReturnValue(IOilFld,record).AsString
  LFile.WriteElt(conn+","+rate+","+temp+","+conc+","+oil)
end

av.delayedrun("osblexe",nil,3)
av.delayedrun("joindata",nil,6)
av.delayedrun("displayemm",nil,8)
'----------------------------------------------
'---------------------END---------------------
'----------------------------------------------
```

```
’----------------------------------------------
’---------Creation Information-----------------
’----------------------------------------------
’
’Name: joindata
’Author:  Cindy How
’Date:    2/24/97
’Revisions:
’1.  3/23/98 by Cindy How
’    -edited to remove old joins.
’---------------------------------------------
’------------Purpose/Description---------------
’---------------------------------------------
’
’This script will import data output from the naUTilus model
’as a .txt file and join it with the branch attribute table.
’----------------------------------------------
’-------------------Input-----------------------
’----------------------------------------------
’
’1.  Attribute table for the theme branches.shp
’2.  Attribute table for the theme nodes.shp
’3.  Output file from the naUTilus model: obrout.txt
’4.  Output file from the naUTilus model: ondout.txt
’----------------------------------------------
’------------------Output-----------------------
’----------------------------------------------
’
’1.  Updated attribute table with the flow rate,
’    concentration, and temperature in each branch.
’
’----------------------------------------------
’----------Set Working Directory----------------
’----------------------------------------------
CurrentDir=FN.GetCWD.AsString
’msgbox.info(Currentdir.AsString,"Your Current Working Directory is")
’
’----------------------------------------------
’---------Getting initial data------------------
’----------------------------------------------
theProject=av.getProject
TheView = av.GetActiveDoc

’----------------------------------------------
’-----------Add Branch data tables--------------
’----------------------------------------------

BrTheme=TheView.FindTheme("obranch.shp")
BrFtab = BrTheme.GetFtab
Field1 = BrFtab.FindField("osbl_")

TabName = "obrout.txt".AsFilename
NewTab = Vtab.Make(TabName,false,false)
Field2 = NewTab.FindField("Branch")

’-----------------------------------------------
’----------Add Node data tables------------------
’-----------------------------------------------

NodeTheme = TheView.FindTheme("onode.shp")
NodeFtab = NodeTheme.GetFtab
NField1 = NodeFtab.FindField("Node")

NTabName = "ondout.txt".AsFilename
NewNTab = Vtab.Make(NtabName,false,false)
Nfield2 = NewNtab.FindField("Node")

’------------------------------------
’----------Remove old Joins------------
’------------------------------------
BrFtab.UnjoinAll
NodeFtab.UnjoinAll

’---------------------------------------------------------
’------Join the attribute table to the naUTilus output-------
’---------------------------------------------------------
’

BrFtab.Join(Field1,NewTab,Field2)
NodeFtab.Join(Nfield1,NewNtab,NField2)

’----------------------------------------------
’---------------------END----------------------
’----------------------------------------------
```

```
'----------------------------------------------
'---------Creation Information-----------------
'----------------------------------------------
'
'Name: displayemm
'Author:  Cindy How
'Date:    3/97
'Revisions:
'1. 3/23/98 by Cindy How
'    - removed option of "absolute emissions"
'
'----------------------------------------------
'------------Purpose/Description---------------
'----------------------------------------------
'
'This script offers two options in displaying the
'data output from naUTilus.
'It the maximum and minimum emissions at
'the nodes, breaks the range into 4 intervals, and creates
'colored circles (coded by emissions range) around the nodes.
'
'-----------------------------------------------
'-------------------Input-----------------------
'-----------------------------------------------
'
'1.  Attribute table for the theme nodes (joined with naUTilus output)
'
'-----------------------------------------------
'------------------Output-----------------------
'-----------------------------------------------
'
'-----------------------------------------------
'---------Get Initial Information---------------
'-----------------------------------------------
TheProject = av.GetProject
TheView = av.GetActiveDoc
TheTheme= TheView.FindTheme("onode.shp")
TheFtab = TheTheme.GetFtab
TheField = TheFtab.FindField("Emissions")
TFld = TheFtab.FindField("Node_type")

'------------------------------------------------
'-----------Setup Legend options-----------------
'------------------------------------------------
```

```
theLegend = Thetheme.GetLegend
SymbList = theLegend.GetSymbols
SColor = Color.GetYellow
EColor = Color.GetRed



theLegend.ResetClasses(theTheme,theField.AsString)
theLegend.SetLegendType(#Legend_type_color)
theLegend.SetPrecision(-13)
theLegend.Interval(theTheme,theField.AsString,4)
'theLegend.Natural(theTheme,theField.AsString,4)
theColorRamp = SymbList.RampColors(SColor,EColor)
theLegend.DisplayNoDataClass(true)
NilSymb = theLegend.GetNullSymbol
NilColor = Color.GetWhite
NilColor.SetTransparent(True)
NilSymb.SetColor(NilColor)

SymbList.RampSizes(6,6)
'For each i in SymbList



 '-----------------------------------------------
 '------------Find Maximum emission--------------
 '-----------------------------------------------
   maxnum = 0
  minnum = 0
  i = 0
  For each record in TheFtab
   nilflag = TheFtab.ReturnValue(TFld,record)
   check = TheFtab.ReturnValue(TheField,record)
   if ((nilflag = -1) or (nilflag = 2)) then
'     msgbox.info("record"++record.AsString++"is null","user test")
   else
     if (maxnum < check) then
       maxnum = check
     else
     end
   '----------------------------------------------
   '-------------Find Minimum emission-------------
   '----------------------------------------------
```

```
  'Set initial minimum to emission value from node 1
   if ( i = 0) then
     minnum = check
   else
   end

   if (minnum > check) then
     minnum = check
   else
   end

   i = i + 1
  end
end

'Check Max/Min values.
'msgBox.info("Minimum ="++minnum.AsString+", Maximum
="++maxnum.AsString,"Minimum and maximum values")


'Divide into 4 ranges
intervalsize = (maxnum - minnum)/4

val = minnum + intervalsize
k = 1
for each int in theLegend.GetClassifications
  cls = int
  if (k > 1) then
    cls.setMinimum(val - intervalsize)
    lval = val - intervalsize
  else
    cls.setMinimum(0)
    lval = 0
  end
  cls.setMaximum(val)
  LString = lval.AsString++"-"++val.asString
  cls.SetLabel(LString)
  val = val + intervalsize
  theTheme.UpdateLegend
  k = k + 1
end

'Set Legend symbol and labels for null values
NullList = {"No emissions - branch end","0 - 0",NilSymb,0,0}
theLegend.SetClassInfo(4,NullList)

'-----------------------------------------------
'---------------------END---------------------
'-----------------------------------------------
```

Appendix D

EXAMPLE NAUTILUS INPUT AND OUTPUT FILES

Example naUTilus ISBL input file: 1,3-butadiene, baseline conditions (isbl.in)

```
no.branches, no.nodes, no. drains, open/closed
1 = open/mixed, 2 = closed
225,136,97,1
last node
44
Henry’s law Constant - Method 1,2,3,4
2
2.92,25,6.8500,930.546,238.854
Liquid Diffusivity, Gas Diffusivity
0.000012,0.108
Ambient Temperature (C) and relative humidity
20,0.5
no. pickholes per cover, pickhole diameter (m)
4,0.0254
Ambient wind velocity (m/s)
1
1 = no oil, 2 = oil only at surface
3 = oil only dispersed, not at surface
1
If oil is present, read Kow, density, oil MW (else, empty)
Indicate method of determining vapor pressure if oil is present
Method 1-Antoine’s constants
Enter A,B,C for T in K giving VP in mmHg
Method 2-Vapor Pressure = constant
Enter Vapor Pressure
1 = Parkhurst&Pomeroy, 2 = Owens-Edwards-Gibbs
1
1 = Mass Transfer, 2 = Equilibrium
1
Enter 1 to accept default reach length
2
User specified branch characteristics
Length, Diameter, slope for each branch
0.488,0.1524,0.06
0.488,0.1524,0.06
2.87,0.1524,0.06
0.244,0.1524,0.06
1.68,0.1524,0.06
3.32,0.1524,0.06
0.488,0.1524,0.06
4.05,0.1524,0.06
0.945,0.1524,0.06
3.11,0.1524,0.06
0.945,0.1524,0.06
1.43,0.1524,0.06
2.13,0.1524,0.06
2.87,0.1524,0.06
6.19,0.1524,0.06
3.11,0.1524,0.06
0.244,0.1524,0.06
1.43,0.1524,0.06
3.32,0.1524,0.06
7.38,0.1524,0.06
1.92,0.1524,0.06
0.701,0.1524,0.06
0.488,0.1524,0.06
1.19,0.1524,0.06
0.244,0.1524,0.06
0.701,0.1524,0.06
1.92,0.1524,0.06
1.68,0.1524,0.06
3.32,0.1524,0.06
0.244,0.1524,0.06
1.68,0.1524,0.06
1.19,0.1524,0.06
0.488,0.1524,0.06
5.7,0.1524,0.06
0.488,0.1524,0.06
1.68,0.1524,0.06
3.11,0.1524,0.06
1.68,0.1524,0.06
0.701,0.1524,0.06
0.701,0.1524,0.06
4.51,0.1524,0.06
0.244,0.1524,0.06
2.62,0.1524,0.06
0.945,0.1524,0.06
1.43,0.1524,0.06
3.32,0.1524,0.06
0.488,0.1524,0.06
1.68,0.1524,0.06
4.05,0.1524,0.06
1.43,0.1524,0.06
0.244,0.1524,0.06
1.43,0.1524,0.06
```

2.13,0.1524,0.06
4.51,0.1524,0.06
9.05,0.1524,0.06
3.57,0.1524,0.06
1.43,0.1524,0.06
4.3,0.1524,0.06
0.701,0.1524,0.06
0.244,0.1524,0.06
6.43,0.1524,0.06
2.62,0.1524,0.06
0.701,0.1524,0.06
2.13,0.1524,0.06
0.945,0.1524,0.06
2.62,0.1524,0.06
0.488,0.1524,0.06
2.13,0.1524,0.06
0.945,0.1524,0.06
3.11,0.1524,0.06
1.68,0.1524,0.06
0.488,0.1524,0.06
0.945,0.1524,0.06
0.945,0.1524,0.06
1.19,0.1524,0.06
1.19,0.1524,0.06
0.701,0.1524,0.06
0.488,0.1524,0.06
0.488,0.1524,0.06
1.43,0.1524,0.06
1.43,0.1524,0.06
1.19,0.1524,0.06
0.945,0.1524,0.06
1.68,0.1524,0.06
1.43,0.1524,0.06
0.488,0.1524,0.06
1.68,0.1524,0.06
0.701,0.1524,0.06
2.87,0.1524,0.06
1.43,0.1524,0.06
1.92,0.1524,0.06
4.75,0.1524,0.06
0.945,0.1524,0.06
5,0.1524,0.06
6.68,0.1524,0.06
1.19,0.1524,0.06
2.87,0.1524,0.06
0.244,0.1524,0.06
5.7,0.4572,0.06
1.43,0.1524,0.06
2.87,0.4572,0.06
18.11,0.4572,0.06
1.19,0.1524,0.06
9.05,0.4572,0.06
8.56,0.4572,0.06
1.43,0.1524,0.06
4.3,0.4572,0.06
8.41,0.4572,0.06
0.701,0.1524,0.06
4.21,0.4572,0.06
11.92,0.4572,0.06
2.87,0.1524,0.06
5.94,0.4572,0.06
8.9,0.4572,0.06
2.62,0.1524,0.06
4.45,0.4572,0.06
8.56,0.1524,0.06
3.57,0.1524,0.06
1.68,0.1524,0.06
2.87,0.1524,0.06
2.38,0.1524,0.06
7.13,0.1524,0.06
2.13,0.1524,0.06
5,0.1524,0.06
0.945,0.1524,0.06
3.57,0.1524,0.06
3.57,0.1524,0.06
7.38,0.1524,0.06
0.488,0.1524,0.06
1.89,0.1524,0.06
0.488,0.1524,0.06
0.488,0.1524,0.06
0.488,0.1524,0.06
0.488,0.1524,0.06
3.32,0.1524,0.06
0.488,0.1524,0.06
1.43,0.1524,0.06
0.488,0.1524,0.06
1.92,0.1524,0.06
0.701,0.1524,0.06

3.32,0.1524,0.06
0.244,0.1524,0.06
1.68,0.1524,0.06
0.488,0.1524,0.06
0.244,0.1524,0.06
1.68,0.1524,0.06
0.244,0.1524,0.06
4.51,0.1524,0.06
0.488,0.1524,0.06
1.92,0.1524,0.06
0.488,0.1524,0.06
0.488,0.1524,0.06
2.62,0.1524,0.06
0.488,0.1524,0.06
0.244,0.1524,0.06
1.19,0.1524,0.06
0.488,0.1524,0.06
1.68,0.1524,0.06
5.24,0.1524,0.06
1.92,0.1524,0.06
0.945,0.1524,0.06
3.57,0.1524,0.06
0.01,0.1524,0.06
0.244,0.1524,0.06
0.244,0.1524,0.06
0.488,0.1524,0.06
2.87,0.1524,0.06
2.87,0.1524,0.06
0.488,0.1524,0.06
0.244,0.1524,0.06
0.488,0.1524,0.06
3.11,0.1524,0.06
8.32,0.1524,0.06
0.488,0.1524,0.06
1.43,0.1524,0.06
0.945,0.1524,0.06
0.945,0.1524,0.06
2.38,0.1524,0.06
0.488,0.1524,0.06
1.89,0.1524,0.06
3.57,0.1524,0.06
5.49,0.1524,0.06
0.488,0.1524,0.06
5.7,0.1524,0.06
1.92,0.1524,0.06
0.488,0.1524,0.06
5.49,0.1524,0.06
5.73,0.1524,0.06
1.68,0.1524,0.06
0.701,0.1524,0.06
0.945,0.1524,0.06
2.87,0.1524,0.06
2.13,0.1524,0.06
3.32,0.1524,0.06
0.945,0.1524,0.06
2.38,0.1524,0.06
3.11,0.1524,0.06
3.57,0.1524,0.06
1.68,0.1524,0.06
0.488,0.1524,0.06
0.701,0.1524,0.06
3.32,0.1524,0.06
1.43,0.1524,0.06
3.32,0.1524,0.06
4.05,0.1524,0.06
0.701,0.1524,0.06
1.43,0.1524,0.06
6.19,0.1524,0.06
1.43,0.1524,0.06
0.701,0.1524,0.06
0.488,0.1524,0.06
1.43,0.1524,0.06
1.19,0.1524,0.06
1.43,0.1524,0.06
2.87,0.1524,0.06
1.43,0.1524,0.06
1.92,0.1524,0.06
4.05,0.1524,0.06
3.32,0.1524,0.06
9.3,0.1524,0.06
1.19,0.1524,0.06
1.19,0.1524,0.06
4.05,0.1524,0.06
0.488,0.1524,0.06
3.57,0.1524,0.06
Enter drain characteristics: type - elbow(1)/on line(2), branch/node connectivity, flowrate (L/s), temperature(C), concentration (mg/L), drain diameter (m),
unseald(1)/sealed(2> drain, oil vol./total vol.

1,1,1,30,5,0.0508,1,0
1,2,0,30,0,0.127,1,0
1,4,0,30,0,0.203,1,0
1,7,0,30,0,0.203,1,0
1,11,0,30,0,0.127,1,0
2,6,0,30,0,0.127,1,0
1,14,0,30,0,0.127,1,0
1,16,0,30,0,0.203,1,0
1,21,0.5,30,1,0.127,1,0
1,22,0,30,0,0.127,1,0
1,23,1,30,0,0.0508,1,0
1,24,0,30,0,0.127,1,0
1,25,0,30,0,0.127,1,0
1,26,0.5,30,0,0.127,1,0
1,27,0,30,0,0.2,1,0
1,30,0,30,0,0.127,1,0
1,31,0,30,0,0.127,1,0
1,32,0,30,0,0.127,1,0
1,34,0,30,0,0.127,1,0
2,15,0.5,30,0,0.0508,1,0
1,37,0,30,0,0.127,1,0
1,39,2,30,0,0.0508,1,0
1,40,1,30,0,0.0508,1,0
1,54,0,30,0,0.127,1,0
2,34,0,30,0,0.2,1,0
1,66,1,30,1,0.0508,1,0
1,67,0,30,0,0.127,1,0
1,69,0,30,0,0.267,1,0
1,71,0,30,0,0.127,1,0
2,36,0,30,0,0.2,1,0
1,74,0,30,0,0.127,1,0
1,75,0,30,0,0.127,1,0
1,76,0,30,0,0.127,1,0
1,77,0,30,0,0.127,1,0
1,78,0,30,0,0.127,1,0
1,79,0,30,0,0.127,1,0
1,80,0,30,0,0.127,1,0
1,81,0,30,0,0.127,1,0
1,82,0,30,0,0.127,1,0
1,83,0,30,0,0.203,1,0
1,84,0,30,0,0.127,1,0
1,86,0,30,0,0.203,1,0
1,87,0,30,0,0.127,1,0
1,88,0,30,0,0.127,1,0
1,89,0,30,0,0.127,1,0
1,96,0,30,0,0.159,1,0
1,98,0,30,0,0.127,1,0
1,120,0,30,0,0.127,1,0
1,121,0,30,0,0.127,1,0
2,57,0.5,30,1,0.0508,1,0
1,125,0,30,0,0.127,1,0
1,129,0,30,0,0.127,1,0
1,131,0,30,0,0.127,1,0
1,132,0,30,0,0.21,1,0
1,134,0,30,0,0.127,1,0
1,136,1,30,1,0.0508,1,0
1,138,0,30,0,0.127,1,0
1,142,0,30,0,0.127,1,0
1,145,0,30,0,0.127,1,0
1,147,0,30,0,0.127,1,0
1,149,0,30,0,0.127,1,0
1,152,0,30,0,0.127,1,0
1,155,0,30,0,0.127,1,0
1,157,0,30,0,0.127,1,0
1,160,0,30,0,0.127,1,0
1,164,0,30,0,0.127,1,0
1,166,0,30,0,0.127,1,0
1,169,0,30,0,0.127,1,0
1,171,0,30,0,0.127,1,0
1,174,0,30,0,0.127,1,0
1,176,0,30,0,0.127,1,0
1,177,0,30,0,0.127,1,0
1,178,0,30,0,0.127,1,0
1,179,0,30,0,0.127,1,0
1,182,0,30,0,0.127,1,0
1,183,0,30,0,0.127,1,0
2,85,0,30,0,0.2,1,0
1,186,0,30,0,0.127,1,0
1,188,0,30,0,0.127,1,0
1,190,0,30,0,0.127,1,0
1,192,0,30,0,0.127,1,0
2,90,0,30,0,0.2,1,0
2,91,0,30,0,0.127,1,0
1,196,0,30,0,0.127,1,0
1,198,0,30,0,0.127,1,0
1,200,0,30,0,0.127,1,0
1,201,0,30,0,0.127,1,0
1,203,0,30,0,0.127,1,0

```
1,205,0,30,0,0.127,1,0
1,209,1,30,0,0.127,1,0
1,212,0,30,0,0.127,1,0
1,214,0,30,0,0.127,1,0
1,216,0,30,0,0.127,1,0
1,218,0.5,30,0,0.0508,1,0
1,221,0,30,0,0.0508,1,0
1,222,0,30,0,0.127,1,0
1,224,0,30,0,0.127,1,0
Enter no. of branches connected to node i, kind of node
(junction/manhole)
...1 for junction, 2 for manhole...
Enter branch no., kind of connection
...1 for inflow, -1 for outflow
3,1
2,1
3,1
5,-1
3,1
4,1
5,1
6,-1
3,1
7,1
8,1
15,-1
2,1
9,1
12,-1
2,1
12,1
19,-1
2,1
14,1
13,-1
3,1
15,1
16,1
17,-1
3,1
18,1
20,1
109,-1
3,1
24,1
25,1
49,-1
3,1
27,1
28,1
55,-1
3,1
29,1
30,1
33,-1
2,1
31,1
46,-1
3,1
33,1
34,1
62,-1
3,1
35,1
36,1
72,-1
2,1
37,1
36,-1
3,1
38,1
39,1
95,-1
3,1
41,1
42,1
44,-1
3,1
44,1
83,1
43,-1
2,1
43,1
85,-1
2,1
76,1
45,-1
3,1
```

45,1
74,1
47,-1
3,1
46,1
47,1
48,-1
3,1
48,1
79,1
50,-1
3,1
49,1
50,1
51,-1
3,1
51,1
78,1
52,-1
3,1
52,1
77,1
53,-1
3,1
54,1
56,1
90,-1
3,1
55,1
57,1
56,-1
3,1
58,1
80,1
57,-1
3,1
59,1
81,1
58,-1
3,1
60,1
61,1
97,-1
3,1
63,1
87,1
61,-1
3,1
62,1
64,1
63,-1
2,1
66,1
65,-1
3,1
69,1
70,1
68,-1
2,1
71,1
73,-1
3,1
72,1
73,1
93,-1
3,1
85,1
86,1
115,-1
3,1
88,1
89,1
91,-1
3,1
90,1
91,1
100,-1
3,1
92,1
94,1
103,-1
3,1
95,1
96,1
118,-1
3,1
97,1
98,1

106,-1
2,1
101,1
99,-1
4,2
100,1
102,1
159,1
101,-1
2,1
104,1
102,-1
4,2
103,1
105,1
128,1
104,-1
2,1
107,1
105,-1
3,2
106,1
108,1
107,-1
2,1
110,1
108,-1
4,2
109,1
111,1
122,1
110,-1
2,1
113,1
111,-1
4,2
112,1
114,1
124,1
113,-1
2,1
116,1
114,-1
3,2
115,1
117,1
116,-1
3,1
119,1
120,1
117,-1
2,1
121,1
123,-1
3,1
123,1
126,1
122,-1
2,1
130,1
124,-1
3,1
125,1
129,1
127,-1
3,1
131,1
140,1
128,-1
2,1
163,1
130,-1
3,1
132,1
133,1
135,-1
3,1
134,1
135,1
137,-1
3,1
136,1
137,1
139,-1
3,1
138,1
139,1
141,-1

3,1
142,1
144,1
143,-1
3,1
146,1
187,1
144,-1
3,1
145,1
148,1
146,-1
3,1
147,1
150,1
148,-1
3,1
149,1
151,1
150,-1
3,1
153,1
178,1
151,-1
3,1
152,1
154,1
153,-1
3,1
156,1
161,1
154,-1
3,1
155,1
158,1
156,-1
3,1
160,1
185,1
159,-1
3,1
174,1
175,1
161,-1

3,1
165,1
180,1
163,-1
3,1
164,1
167,1
165,-1
3,1
166,1
168,1
167,-1
3,1
170,1
184,1
168,-1
3,1
169,1
172,1
170,-1
3,1
171,1
173,1
172,-1
3,1
181,1
189,1
180,-1
2,1
194,1
184,-1
3,1
186,1
199,1
185,-1
3,1
188,1
193,1
187,-1
3,1
190,1
191,1
189,-1
3,1

192,1
202,1
191,-1
2,1
197,1
193,-1
2,1
196,1
195,-1
3,1
200,1
206,1
199,-1
4,1
201,1
203,1
204,1
202,-1
3,1
205,1
219,1
204,-1
3,1
208,1
222,1
207,-1
3,1
210,1
223,1
208,-1
3,1
209,1
211,1
210,-1
3,1
213,1
221,1
211,-1
3,1
212,1
215,1
213,-1
3,1
214,1
217,1
215,-1
2,1
1,1
9,-1
2,1
10,1
3,-1
2,1
6,1
8,-1
2,1
11,1
10,-1
2,1
13,1
18,-1
2,1
17,1
60,-1
2,1
19,1
20,-1
2,1
21,1
28,-1
2,1
22,1
41,-1
2,1
23,1
29,-1
2,1
26,1
38,-1
2,1
32,1
35,-1
2,1
40,1
42,-1
3,1
53,1
75,1

92,-1
2,1
82,1
59,-1
2,1
84,1
64,-1
3,1
65,1
68,1
112,-1
2,1
67,1
70,-1
2,1
93,1
94,-1
2,1
118,1
119,-1
2,1
127,1
126,-1
2,1
162,1
133,-1
3,1
141,1
143,1
140,-1
2,1
157,1
158,-1
3,1
176,1
177,1
162,-1
2,1
182,1
173,-1
2,1
179,1
175,-1
2,1
183,1
181,-1
2,1
195,1
194,-1
2,1
198,1
197,-1
2,1
207,1
206,-1
2,1
216,1
217,-1
2,1
218,1
220,-1
2,1
220,1
219,-1
2,1
225,1
223,-1
2,1
224,1
225,-1
Number of drops
10
enter node, branch, drop height (m), tailwater depth (m) for each drop
45,100,0.1524,0.6096
45,159,0.1524,0.6096
47,103,0.1524,0.6096
47,128,0.1524,0.6096
49,106,0.1524,0.6096
51,109,0.1524,0.6096
51,122,0.1524,0.6096
53,112,0.1524,0.6096
53,124,0.1524,0.6096
55,115,0.1524,0.6096
Enter no. of hard-piped connections
0
Enter node connectivity, flow rate (L/s), temperature (C),
concentration (mg/L), oil volume/total volume

Example naUTilus output file: 1,3-butadiene, baseline conditions (isbl.out)

```
---------------------------------------------
″
naUTilus
″

″
A Model for Predicting Chemical
Emissions from Industrial Sewers
version 1.0

developed by
David A. Olson, Sunil Varma, Richard L. Corsi

Program in Air Resources Engineering
The University of Texas at Austin

---------------------------------------------
```


Network water flow rates for each branch (L/s)

```
qibr(   1 )=    1.000
qibr(   2 )=     .000
qibr(   3 )=     .000
qibr(   4 )=     .000
qibr(   5 )=     .000
qibr(   6 )=     .000
qibr(   7 )=     .000
qibr(   8 )=     .000
qibr(   9 )=    1.000
qibr(  10 )=     .000
qibr(  11 )=     .000
qibr(  12 )=    1.000
qibr(  13 )=     .000
qibr(  14 )=     .000
qibr(  15 )=     .000
qibr(  16 )=     .000
qibr(  17 )=     .000
qibr(  18 )=     .000
qibr(  19 )=    1.000
qibr(  20 )=    1.000
qibr(  21 )=     .500
qibr(  22 )=     .000
qibr(  23 )=    1.000
qibr(  24 )=     .000
qibr(  25 )=     .000
qibr(  26 )=     .500
qibr(  27 )=     .000
qibr(  28 )=     .500
qibr(  29 )=    1.000
qibr(  30 )=     .000
qibr(  31 )=     .000
qibr(  32 )=     .000
qibr(  33 )=    1.000
qibr(  34 )=     .000
qibr(  35 )=     .000
qibr(  36 )=     .500
qibr(  37 )=     .000
qibr(  38 )=     .500
qibr(  39 )=    2.000
qibr(  40 )=    1.000
qibr(  41 )=     .000
qibr(  42 )=    1.000
qibr(  43 )=    1.000
qibr(  44 )=    1.000
qibr(  45 )=     .000
qibr(  46 )=     .000
qibr(  47 )=     .000
qibr(  48 )=     .000
qibr(  49 )=     .000
qibr(  50 )=     .000
qibr(  51 )=     .000
qibr(  52 )=     .000
qibr(  53 )=     .000
qibr(  54 )=     .000
qibr(  55 )=     .500
qibr(  56 )=     .500
qibr(  57 )=     .000
qibr(  58 )=     .000
qibr(  59 )=     .000
qibr(  60 )=     .000
qibr(  61 )=    1.000
qibr(  62 )=    1.000
qibr(  63 )=    1.000
```

qibr(64)= .000
qibr(65)= 1.000
qibr(66)= 1.000
qibr(67)= .000
qibr(68)= .000
qibr(69)= .000
qibr(70)= .000
qibr(71)= .000
qibr(72)= .500
qibr(73)= .000
qibr(74)= .000
qibr(75)= .000
qibr(76)= .000
qibr(77)= .000
qibr(78)= .000
qibr(79)= .000
qibr(80)= .000
qibr(81)= .000
qibr(82)= .000
qibr(83)= .000
qibr(84)= .000
qibr(85)= 1.000
qibr(86)= .000
qibr(87)= .000
qibr(88)= .000
qibr(89)= .000
qibr(90)= .500
qibr(91)= .000
qibr(92)= .000
qibr(93)= .500
qibr(94)= .500
qibr(95)= 2.500
qibr(96)= .000
qibr(97)= 1.000
qibr(98)= .000
qibr(99)= 10.500
qibr(100)= .500
qibr(101)= 10.500
qibr(102)= 9.000
qibr(103)= .500
qibr(104)= 9.000
qibr(105)= 7.500
qibr(106)= 1.000
qibr(107)= 7.500
qibr(108)= 6.500
qibr(109)= 1.000
qibr(110)= 6.500
qibr(111)= 5.000
qibr(112)= 1.000
qibr(113)= 5.000
qibr(114)= 3.500
qibr(115)= 1.000
qibr(116)= 3.500
qibr(117)= 2.500
qibr(118)= 2.500
qibr(119)= 2.500
qibr(120)= .000
qibr(121)= .000
qibr(122)= .500
qibr(123)= .500
qibr(124)= .500
qibr(125)= .000
qibr(126)= .000
qibr(127)= .000
qibr(128)= 1.000
qibr(129)= .000
qibr(130)= .500
qibr(131)= .000
qibr(132)= .000
qibr(133)= .000
qibr(134)= .000
qibr(135)= .000
qibr(136)= 1.000
qibr(137)= .000
qibr(138)= .000
qibr(139)= 1.000
qibr(140)= 1.000
qibr(141)= 1.000
qibr(142)= .000
qibr(143)= .000
qibr(144)= .000
qibr(145)= .000
qibr(146)= .000
qibr(147)= .000
qibr(148)= .000
qibr(149)= .000
qibr(150)= .000
qibr(151)= .000

qibr(152)= .000
qibr(153)= .000
qibr(154)= .000
qibr(155)= .000
qibr(156)= .000
qibr(157)= .000
qibr(158)= .000
qibr(159)= 1.000
qibr(160)= .000
qibr(161)= .000
qibr(162)= .000
qibr(163)= .500
qibr(164)= .000
qibr(165)= .000
qibr(166)= .000
qibr(167)= .000
qibr(168)= .000
qibr(169)= .000
qibr(170)= .000
qibr(171)= .000
qibr(172)= .000
qibr(173)= .000
qibr(174)= .000
qibr(175)= .000
qibr(176)= .000
qibr(177)= .000
qibr(178)= .000
qibr(179)= .000
qibr(180)= .500
qibr(181)= .000
qibr(182)= .000
qibr(183)= .000
qibr(184)= .000
qibr(185)= 1.000
qibr(186)= .000
qibr(187)= .000
qibr(188)= .000
qibr(189)= .500
qibr(190)= .000
qibr(191)= .500
qibr(192)= .000
qibr(193)= .000
qibr(194)= .000
qibr(195)= .000
qibr(196)= .000
qibr(197)= .000
qibr(198)= .000
qibr(199)= 1.000
qibr(200)= .000
qibr(201)= .000
qibr(202)= .500
qibr(203)= .000
qibr(204)= .500
qibr(205)= .000
qibr(206)= 1.000
qibr(207)= 1.000
qibr(208)= 1.000
qibr(209)= 1.000
qibr(210)= 1.000
qibr(211)= .000
qibr(212)= .000
qibr(213)= .000
qibr(214)= .000
qibr(215)= .000
qibr(216)= .000
qibr(217)= .000
qibr(218)= .500
qibr(219)= .500
qibr(220)= .500
qibr(221)= .000
qibr(222)= .000
qibr(223)= .000
qibr(224)= .000
qibr(225)= .000

Network water temperatures for each branch (C)

tibr(1)= 30.00
tibr(2)= 30.00
tibr(3)= 20.00
tibr(4)= 30.00
tibr(5)= 20.00
tibr(6)= 20.00
tibr(7)= 30.00
tibr(8)= 20.00
tibr(9)= 30.00
tibr(10)= 20.00
tibr(11)= 30.00

tibr(12)= 30.00
tibr(13)= 20.00
tibr(14)= 30.00
tibr(15)= 20.00
tibr(16)= 30.00
tibr(17)= 20.00
tibr(18)= 20.00
tibr(19)= 30.00
tibr(20)= 30.00
tibr(21)= 30.00
tibr(22)= 30.00
tibr(23)= 30.00
tibr(24)= 30.00
tibr(25)= 30.00
tibr(26)= 30.00
tibr(27)= 30.00
tibr(28)= 30.00
tibr(29)= 30.00
tibr(30)= 30.00
tibr(31)= 30.00
tibr(32)= 30.00
tibr(33)= 30.00
tibr(34)= 30.00
tibr(35)= 20.00
tibr(36)= 30.00
tibr(37)= 30.00
tibr(38)= 30.00
tibr(39)= 30.00
tibr(40)= 30.00
tibr(41)= 20.00
tibr(42)= 30.00
tibr(43)= 30.00
tibr(44)= 30.00
tibr(45)= 20.00
tibr(46)= 20.00
tibr(47)= 20.00
tibr(48)= 20.00
tibr(49)= 20.00
tibr(50)= 20.00
tibr(51)= 20.00
tibr(52)= 20.00
tibr(53)= 20.00
tibr(54)= 30.00
tibr(55)= 30.00
tibr(56)= 30.00
tibr(57)= 20.00
tibr(58)= 20.00
tibr(59)= 20.00
tibr(60)= 20.00
tibr(61)= 30.00
tibr(62)= 30.00
tibr(63)= 30.00
tibr(64)= 20.00
tibr(65)= 30.00
tibr(66)= 30.00
tibr(67)= 30.00
tibr(68)= 20.00
tibr(69)= 30.00
tibr(70)= 20.00
tibr(71)= 30.00
tibr(72)= 30.00
tibr(73)= 20.00
tibr(74)= 30.00
tibr(75)= 30.00
tibr(76)= 30.00
tibr(77)= 30.00
tibr(78)= 30.00
tibr(79)= 30.00
tibr(80)= 30.00
tibr(81)= 30.00
tibr(82)= 30.00
tibr(83)= 30.00
tibr(84)= 30.00
tibr(85)= 30.00
tibr(86)= 30.00
tibr(87)= 30.00
tibr(88)= 30.00
tibr(89)= 30.00
tibr(90)= 30.00
tibr(91)= 20.00
tibr(92)= 20.00
tibr(93)= 30.00
tibr(94)= 30.00
tibr(95)= 30.00
tibr(96)= 30.00
tibr(97)= 30.00
tibr(98)= 30.00
tibr(99)= 30.00

tibr(100)=	30.00
tibr(101)=	30.00
tibr(102)=	30.00
tibr(103)=	30.00
tibr(104)=	30.00
tibr(105)=	30.00
tibr(106)=	30.00
tibr(107)=	30.00
tibr(108)=	30.00
tibr(109)=	30.00
tibr(110)=	30.00
tibr(111)=	30.00
tibr(112)=	30.00
tibr(113)=	30.00
tibr(114)=	30.00
tibr(115)=	30.00
tibr(116)=	30.00
tibr(117)=	30.00
tibr(118)=	30.00
tibr(119)=	30.00
tibr(120)=	30.00
tibr(121)=	30.00
tibr(122)=	30.00
tibr(123)=	30.00
tibr(124)=	30.00
tibr(125)=	30.00
tibr(126)=	20.00
tibr(127)=	20.00
tibr(128)=	30.00
tibr(129)=	30.00
tibr(130)=	30.00
tibr(131)=	30.00
tibr(132)=	30.00
tibr(133)=	20.00
tibr(134)=	30.00
tibr(135)=	20.00
tibr(136)=	30.00
tibr(137)=	20.00
tibr(138)=	30.00
tibr(139)=	30.00
tibr(140)=	30.00
tibr(141)=	30.00
tibr(142)=	30.00
tibr(143)=	20.00
tibr(144)=	20.00
tibr(145)=	30.00
tibr(146)=	20.00
tibr(147)=	30.00
tibr(148)=	20.00
tibr(149)=	30.00
tibr(150)=	20.00
tibr(151)=	20.00
tibr(152)=	30.00
tibr(153)=	20.00
tibr(154)=	20.00
tibr(155)=	30.00
tibr(156)=	20.00
tibr(157)=	30.00
tibr(158)=	20.00
tibr(159)=	30.00
tibr(160)=	30.00
tibr(161)=	20.00
tibr(162)=	20.00
tibr(163)=	30.00
tibr(164)=	30.00
tibr(165)=	20.00
tibr(166)=	30.00
tibr(167)=	20.00
tibr(168)=	20.00
tibr(169)=	30.00
tibr(170)=	20.00
tibr(171)=	30.00
tibr(172)=	20.00
tibr(173)=	20.00
tibr(174)=	30.00
tibr(175)=	20.00
tibr(176)=	30.00
tibr(177)=	30.00
tibr(178)=	30.00
tibr(179)=	30.00
tibr(180)=	30.00
tibr(181)=	20.00
tibr(182)=	30.00
tibr(183)=	30.00
tibr(184)=	20.00
tibr(185)=	30.00
tibr(186)=	30.00
tibr(187)=	20.00

tibr(188)= 30.00
tibr(189)= 30.00
tibr(190)= 30.00
tibr(191)= 30.00
tibr(192)= 30.00
tibr(193)= 20.00
tibr(194)= 20.00
tibr(195)= 20.00
tibr(196)= 30.00
tibr(197)= 20.00
tibr(198)= 30.00
tibr(199)= 30.00
tibr(200)= 30.00
tibr(201)= 30.00
tibr(202)= 30.00
tibr(203)= 30.00
tibr(204)= 30.00
tibr(205)= 30.00
tibr(206)= 30.00
tibr(207)= 30.00
tibr(208)= 30.00
tibr(209)= 30.00
tibr(210)= 30.00
tibr(211)= 20.00
tibr(212)= 30.00
tibr(213)= 20.00
tibr(214)= 30.00
tibr(215)= 20.00
tibr(216)= 30.00
tibr(217)= 20.00
tibr(218)= 30.00
tibr(219)= 30.00
tibr(220)= 30.00
tibr(221)= 30.00
tibr(222)= 30.00
tibr(223)= 20.00
tibr(224)= 30.00
tibr(225)= 20.00

Network air flows for each branch (L/s)

qair(1)= 1.736
qair(2)= .450
qair(3)= .450
qair(4)= .450
qair(5)= .899
qair(6)= 1.349
qair(7)= .450
qair(8)= 1.349
qair(9)= 1.736
qair(10)= .450
qair(11)= .450
qair(12)= 1.736
qair(13)= .899
qair(14)= .450
qair(15)= 1.799
qair(16)= .450
qair(17)= 2.248
qair(18)= .899
qair(19)= 1.736
qair(20)= 1.736
qair(21)= 1.736
qair(22)= .450
qair(23)= 1.736
qair(24)= .450
qair(25)= .450
qair(26)= 1.736
qair(27)= .450
qair(28)= 1.736
qair(29)= 1.736
qair(30)= .450
qair(31)= .450
qair(32)= .450
qair(33)= 2.186
qair(34)= .450
qair(35)= .450
qair(36)= 2.186
qair(37)= .450
qair(38)= 1.736
qair(39)= 1.736
qair(40)= 1.736
qair(41)= .450
qair(42)= 1.736
qair(43)= 2.636
qair(44)= 2.186
qair(45)= .450
qair(46)= .450
qair(47)= .899

qair(48)= 1.349
qair(49)= .899
qair(50)= 1.799
qair(51)= 2.698
qair(52)= 3.147
qair(53)= 3.597
qair(54)= .450
qair(55)= 2.186
qair(56)= 3.535
qair(57)= 1.349
qair(58)= .899
qair(59)= .450
qair(60)= 2.248
qair(61)= 3.535
qair(62)= 2.636
qair(63)= 3.085
qair(64)= .450
qair(65)= .899
qair(66)= 1.736
qair(67)= .450
qair(68)= .899
qair(69)= .450
qair(70)= .450
qair(71)= .450
qair(72)= 2.636
qair(73)= .899
qair(74)= .450
qair(75)= .450
qair(76)= .450
qair(77)= .450
qair(78)= .450
qair(79)= .450
qair(80)= .450
qair(81)= .450
qair(82)= .450
qair(83)= .450
qair(84)= .450
qair(85)= 2.636
qair(86)= .450
qair(87)= .450
qair(88)= .450
qair(89)= .450
qair(90)= 3.985
qair(91)= .899
qair(92)= 4.047
qair(93)= 3.535
qair(94)= 3.535
qair(95)= 3.473
qair(96)= .450
qair(97)= 5.783
qair(98)= .450
qair(99)= 43.615
qair(100)= 4.884
qair(101)= 43.615
qair(102)= 35.521
qair(103)= 7.582
qair(104)= 35.521
qair(105)= 20.234
qair(106)= 6.233
qair(107)= 20.234
qair(108)= 15.288
qair(109)= 2.636
qair(110)= 15.288
qair(111)= 12.140
qair(112)= 1.799
qair(113)= 12.140
qair(114)= 3.597
qair(115)= 3.085
qair(116)= 3.597
qair(117)= 4.372
qair(118)= 3.923
qair(119)= 3.923
qair(120)= .450
qair(121)= .450
qair(122)= 3.085
qair(123)= 2.186
qair(124)= 8.031
qair(125)= .450
qair(126)= .899
qair(127)= .899
qair(128)= 10.280
qair(129)= .450
qair(130)= 8.031
qair(131)= .450
qair(132)= .450
qair(133)= .899
qair(134)= .450
qair(135)= 1.349

qair(136)= 1.736
qair(137)= 1.799
qair(138)= .450
qair(139)= 3.535
qair(140)= 9.830
qair(141)= 3.985
qair(142)= .450
qair(143)= 5.845
qair(144)= 5.396
qair(145)= .450
qair(146)= 4.047
qair(147)= .450
qair(148)= 3.597
qair(149)= .450
qair(150)= 3.147
qair(151)= 2.698
qair(152)= .450
qair(153)= 2.248
qair(154)= 1.799
qair(155)= .450
qair(156)= .899
qair(157)= .450
qair(158)= .450
qair(159)= 5.783
qair(160)= .450
qair(161)= .899
qair(162)= .899
qair(163)= 8.031
qair(164)= .450
qair(165)= 3.597
qair(166)= .450
qair(167)= 3.147
qair(168)= 2.698
qair(169)= .450
qair(170)= 1.349
qair(171)= .450
qair(172)= .899
qair(173)= .450
qair(174)= .450
qair(175)= .450
qair(176)= .450
qair(177)= .450
qair(178)= .450
qair(179)= .450
qair(180)= 4.434
qair(181)= .450
qair(182)= .450
qair(183)= .450
qair(184)= 1.349
qair(185)= 5.334
qair(186)= .450
qair(187)= 1.349
qair(188)= .450
qair(189)= 3.985
qair(190)= .450
qair(191)= 3.535
qair(192)= .450
qair(193)= .899
qair(194)= .899
qair(195)= .899
qair(196)= .450
qair(197)= .450
qair(198)= .450
qair(199)= 4.884
qair(200)= .450
qair(201)= .450
qair(202)= 3.085
qair(203)= .450
qair(204)= 2.186
qair(205)= .450
qair(206)= 4.434
qair(207)= 4.434
qair(208)= 3.985
qair(209)= 1.736
qair(210)= 3.535
qair(211)= 1.799
qair(212)= .450
qair(213)= 1.349
qair(214)= .450
qair(215)= .899
qair(216)= .450
qair(217)= .450
qair(218)= 1.736
qair(219)= 1.736
qair(220)= 1.736
qair(221)= .450
qair(222)= .450
qair(223)= .450

qair(224)= .450
qair(225)= .450

Liquid and gas concentrations for each branch (mg/L)

1 Cli 4.7990 Clo 4.7490 eta 1.041%
Cai .1158 Cao .1445

2 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

3 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

4 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

5 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

6 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

7 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

8 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

9 Cli 4.7490 Clo 4.6541 eta 1.999%
Cai .1445 Cao .1992

10 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

11 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

12 Cli 4.6541 Clo 4.5147 eta 2.995%
Cai .1992 Cao .2795

13 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

14 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

15 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

16 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

17 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

18 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

19 Cli 4.5147 Clo 4.2103 eta 6.744%
Cai .2795 Cao .4548

20 Cli 4.2103 Clo 3.6199 eta 14.022%
Cai .4548 Cao .7948

21 Cli .9014 Clo .8475 eta 5.979%
Cai .0284 Cao .0439

22 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

23 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

24 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

25 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

26 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

27 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

28 Cli .8475 Clo .8033 eta 5.219%
Cai .0439 Cao .0566

29 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

30 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

31 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

32 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

33 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

34 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

35 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

36 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

37 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

38 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

39 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

40 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

41 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

42 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

43 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

44 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

45 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

46 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

47 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

48 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

49 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

50 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

51 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

52 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

53 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

54 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

55 Cli .8033 Clo .6042 eta 24.787%
Cai .0450 Cao .0905

56 Cli .6042 Clo .5401 eta 10.610%
Cai .0560 Cao .0651

57 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

58 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

59 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

60 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

61 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

62 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

63 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

64 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

65 Cli .9076 Clo .8900 eta 1.941%
Cai .1027 Cao .1223

66 Cli .9598 Clo .9076 eta 5.438%
Cai .0232 Cao .0532

67 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

68 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

69 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

70 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

71 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

72 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

73 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

74 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

75 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

76 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

77 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

78 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

79 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

80 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

81 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

82 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

83 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

84 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

85 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

86 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

87 Cli .0000 Clo .0000 eta .000%

Cai .0000 Cao .0000

88 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

89 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

90 Cli .5401 Clo .5164 eta 4.385%
Cai .0577 Cao .0607

91 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

92 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

93 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

94 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

95 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

96 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

97 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

98 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

99 Cli .3382 Clo .3242 eta 4.138%
Cai .1020 Cao .1054

100 Cli .5164 Clo .4880 eta 5.494%
Cai .0495 Cao .0524

101 Cli .3455 Clo .3382 eta 2.119%
Cai .1002 Cao .1020

102 Cli .4358 Clo .3760 eta 13.718%
Cai .1007 Cao .1159

103 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

104 Cli .4698 Clo .4358 eta 7.245%
Cai .0921 Cao .1007

105 Cli .5078 Clo .4700 eta 7.437%
Cai .1330 Cao .1470

106 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

107 Cli .5282 Clo .5078 eta 3.859%
Cai .1255 Cao .1330

108 Cli .6627 Clo .6094 eta 8.035%
Cai .1434 Cao .1661

109 Cli 3.6199 Clo 3.5176 eta 2.827%
Cai .5236 Cao .5625

110 Cli .6915 Clo .6627 eta 4.166%
Cai .1312 Cao .1434

111 Cli .1536 Clo .1331 eta 13.347%
Cai .0191 Cao .0276

112 Cli .8900 Clo .8259 eta 7.203%
Cai .0612 Cao .0968

113 Cli .1652 Clo .1536 eta 7.028%
Cai .0143 Cao .0191

114 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

115 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

116 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

117 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

118 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

119 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

120 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

121 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

122 Cli .7911 Clo .6233 eta 21.217%
Cai .0338 Cao .0610

123 Cli .8470 Clo .7911 eta 6.592%
Cai .0350 Cao .0478

124 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

125 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

126 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

127 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

128 Cli .8363 Clo .7034 eta 15.885%
Cai .0159 Cao .0289

129 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

130 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

131 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

132 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

133 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

134 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

135 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

136 Cli .9598 Clo .9498 eta 1.041%
Cai .0232 Cao .0289

137 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

138 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

139 Cli .9498 Clo .9114 eta 4.041%
Cai .0142 Cao .0251

140 Cli .8490 Clo .8363 eta 1.496%
Cai .0154 Cao .0167

141 Cli .9114 Clo .8490 eta 6.855%
Cai .0222 Cao .0379

142 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

143 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

144 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

145 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

146 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

147 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

148 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

149 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

150 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

151 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

152 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

153 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

154 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

155 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

156 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

157 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

158 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

159 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

160 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

161 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

162 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

163 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

164 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

165 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

166 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

167 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

168 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

169 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

170 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

171 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

172 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

173 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

174 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

175 Cli .0000 Clo .0000 eta .000%

Cai .0000 Cao .0000

176 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

177 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

178 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

179 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

180 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

181 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

182 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

183 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

184 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

185 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

186 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

187 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

188 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

189 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

190 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

191 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

192 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

193 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

194 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

195 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

196 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

197 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

198 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

199 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

200 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

201 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

202 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

203 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

204 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

```
205  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

206  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

207  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

208  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

209  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

210  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

211  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

212  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

213  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

214  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

215  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

216  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

217  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

218  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

219  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

220  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

221  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

222  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

223  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

224  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

225  Cli   .0000   Clo   .0000    eta   .000%
     Cai   .0000   Cao   .0000

---------------------------------------------

 Network totals

   Above sewer emissions=    .000 mg/s

   Above sewer emissions (%)=    .00
   Overall stripping efficiency(%)=  57.45
   Total mass entering the system =   8.00  mg/s
   Emission rate=    4.60  mg/s

 ---------------------------------------------
```

Example naUTilus output file: 1,3-butadiene, baseline conditions (isblout.txt)

```
Above_sew_emm,Tot_mass_in,Emission_rate,Flow_rate,    Liq_conc,Temp,Oil_frac

.00000,   8.000,   4.59567,  10.500,  .32422,   30.00,  .00000
```

Example naUTilus OSBL input file
(osbl.in)

```
Number of branches
″
110
″
Number of Nodes
″
83
″
Last branch, Last node
″
110,83
″
Number of manhole covers, number of openings per manhole, opening
area (m2)
″
68,4,5.1e-4
″
Number of junctions with no manholes
″
15
″
Enter node number for junctions with no manholes
12
15
16
27
31
34
42
58
59
65
72
73
74
77
81
Ambient wind velocity (m/s)
1
Ambient Temperature (C) and relative humidity
20,0.5
Henry's law Constant - Method 1,2,3,4
1
5.534,3194
Liquid Diffusivity, Gas Diffusivity
1.0e-5,0.09
1 = no oil, 2 = oil only at surface
3 = oil only dispersed, not at surface
1
If oil is present, read Kow, density, oil MW (else, empty)
Indicate method of determining vapor pressure if oil is present
Method 1-Antoine's constants
Enter A,B,C for T in K giving VP in mmHg
Method 2-Vapor Pressure = constant
Enter Vapor Pressure
Enter 1 to change default reach characteristics
1
Number of branches to be modified
Number, Diameter, Length, slope for each branch
110
1,0.5,11.86,0.01
2,1,85.707,0.01
3,1,79.7604,0.01
4,1,156.09,0.01
5,1,40.6801,0.01
6,1,20.336,0.01
7,1,83.1884,0.01
8,0.5,48.7279,0.01
9,0.5,71.5405,0.01
10,0.5,51.3248,0.01
11,0.5,10.2551,0.01
12,0.5,117.925,0.01
13,0.5,54.3935,0.01
14,1,111.655,0.01
15,0.5,204.336,0.01
16,0.5,14.2955,0.01
17,1,21.0393,0.01
18,0.5,24.546,0.01
19,0.5,147.93,0.01
20,0.5,28.1371,0.01
21,0.5,60.794,0.01
22,0.5,45.7153,0.01
23,0.5,14.9477,0.01
24,0.5,25.9256,0.01
25,0.5,9.028,0.01
```

```
26,0.5,14.3369,0.01
27,0.5,36.4195,0.01
28,0.5,54.589,0.01
29,0.5,101.193,0.01
30,0.5,16.682,0.01
31,0.5,120.736,0.01
32,0.5,34.8988,0.01
33,0.5,59.2733,0.01
34,0.5,34.2601,0.01
35,1,103.17,0.01
36,0.5,215.062,0.01
37,0.5,57.2446,0.01
38,0.5,96.3571,0.01
39,0.5,51.1106,0.01
40,0.5,51.3647,0.01
41,0.5,57.989,0.01
42,0.5,59.638,0.01
43,0.5,12.0507,0.01
44,0.5,26.8188,0.01
45,0.5,45.2094,0.01
46,0.5,65.0557,0.01
47,0.5,12.9328,0.01
48,0.5,58.7115,0.01
49,0.5,61.4578,0.01
50,1,124.81,0.01
51,1,202.914,0.01
52,1,173.999,0.01
53,1,94.7452,0.01
54,1,59.8884,0.01
55,0.5,141.366,0.01
56,0.5,59.8594,0.01
57,1,240.866,0.01
58,0.5,57.8576,0.01
59,1,19.3082,0.01
60,1,77.6319,0.01
61,1,149.087,0.01
62,1,55.4387,0.01
63,1,55.7948,0.01
64,1,41.7901,0.01
65,0.5,9.1892,0.01
66,1,58.9896,0.01
67,1,69.0427,0.01
68,1,68.5839,0.01
69,1,145.685,0.01
70,1,183.294,0.01
71,1,86.5928,0.01
72,1,63.055,0.01
73,0.5,95.6939,0.01
74,0.5,12.8479,0.01
75,0.5,61.214,0.01
76,0.5,78.9904,0.01
77,0.5,58.4997,0.01
78,0.5,68.398,0.01
79,1,139.915,0.01
80,0.5,62.2329,0.01
81,0.5,60.1057,0.01
82,1,116.924,0.01
83,0.5,121.033,0.01
84,0.5,54.96,0.01
85,1,115.287,0.01
86,1,117.264,0.01
87,1,45.4808,0.01
88,0.5,11.907,0.01
89,0.5,64.6871,0.01
90,0.5,69.4781,0.01
91,0.5,48.2838,0.01
92,1,49.1764,0.01
93,0.5,52.3283,0.01
94,1,101.73,0.01
95,0.5,43.9394,0.01
96,0.5,88.8249,0.01
97,0.5,17.773,0.01
98,0.5,93.8139,0.01
99,0.5,29.2427,0.01
100,1,59.1421,0.01
101,0.5,148.048,0.01
102,0.5,14.4263,0.01
103,0.5,36.0077,0.01
104,0.5,29.8517,0.01
105,1,52.9934,0.01
106,0.5,48.0393,0.01
107,0.5,98.7488,0.01
108,1,44.4923,0.01
109,0.5,81.6139,0.01
110,1,200.027,0.01
1 = Parkhurst&Pomeroy, 2 = Owens-Edwards-Gibbs
1
Enter no. of branches connected to node i
```

Branch number, kind of connection
Enter no. of branches connected to node i + 1, etc...
Branch number, kind of connection
...1 for inflow, -1 for outflow
2
1,1
2,-1
2
2,1
3,-1
2
3,1
4,-1
2
4,1
5,-1
2
5,1
6,-1
2
6,1
7,-1
2
7,1
14,-1
3
8,1
29,1
10,-1
3
9,1
10,1
31,-1
2
11,1
12,-1
2
12,1
13,-1
2
13,1
18,-1
3
14,1
15,1
17,-1
2
16,1
15,-1
3
17,1
20,1
35,-1
2
18,1
19,-1
2
19,1
21,-1
2
21,1
20,-1
2
23,1
22,-1
2
22,1
27,-1
2
24,1
23,-1
2
25,1
24,-1
2
26,1
33,-1
2
27,1
28,-1
2
28,1
39,-1
2
30,1
29,-1
2
31,1

32,-1
2
32,1
45,-1
3
33,1
46,1
34,-1
2
34,1
37,-1
3
35,1
36,1
53,-1
2
38,1
36,-1
2
37,1
38,-1
3
39,1
40,1
44,-1
2
41,1
40,-1
2
42,1
41,-1
2
43,1
42,-1
2
44,1
49,-1
2
45,1
48,-1
2
47,1
46,-1
2

48,1
50,-1
3
49,1
50,1
51,-1
2
51,1
52,-1
2
52,1
54,-1
4
53,1
54,1
55,1
57,-1
2
56,1
58,-1
4
57,1
58,1
59,1
71,-1
2
60,1
59,-1
2
63,1
60,-1
2
61,1
62,-1
2
64,1
61,-1
2
62,1
63,-1
2
65,1
64,-1
2

66,1
67,-1
2
67,1
68,-1
2
68,1
69,-1
3
69,1
76,1
70,-1
3
70,1
75,1
72,-1
4
71,1
72,1
73,1
79,-1
2
74,1
73,-1
2
77,1
78,-1
2
78,1
80,-1
3
79,1
80,1
82,-1
2
81,1
83,-1
4
82,1
83,1
84,1
85,-1
2
85,1
86,-1
2
86,1
87,-1
2
87,1
92,-1
2
88,1
89,-1
2
89,1
90,-1
2
90,1
91,-1
2
91,1
93,-1
3
92,1
93,1
94,-1
4
94,1
95,1
96,1
100,-1
2
97,1
98,-1
2
98,1
99,-1
2
99,1
103,-1
3
100,1
101,1
105,-1
2
102,1
101,-1

```
2
103,1
104,-1
2
104,1
106,-1
4
105,1
106,1
107,1
108,-1
3
108,1
109,1
110,-1
Number of drops, Nakasone(1)/WATER8(2)
enter node, branch, drop height (m), tailwater depth (m) for each drop
0,1
Number of nodes at which flow enters the system
13
Enter Branch Number,
Flow rate (L/s), Temperature (C), Concentration (mg/L), and oil fraction
by volume
1,15.1,35,0,0
16,15.1,35,0,0
88,10.1,35,0,0
56,15.1,35,0,0
81,15.1,35,0,0
84,15.1,35,0,0
97,20.1,35,0,0
76,20.1,35,0,0
26,15.1,35,0,0
8,8.5,30.59,0.33442,0
66,8.5,30.59,0.58704,0
47,8.5,30.59,0.34734,0
65,8.5,30.59,0.55368,0
```


Example naUTilus OSBL output file
(osbl.out)

```
---------------------------------------------

                    naUTilus

        A Model for Predicting Chemical
        Emissions from Industrial Sewers
                  version 1.0


                  developed by

  David A. Olson, Sunil Varma, Richard L. Corsi

        Program in Air Resources Engineering
        The University of Texas at Austin

---------------------------------------------


 Network water flows (L/s)

Qbr(  1) =  15.100
Qbr(  2) =  15.100
Qbr(  3) =  15.100
Qbr(  4) =  15.100
Qbr(  5) =  15.100
Qbr(  6) =  15.100
Qbr(  7) =  15.100
Qbr(  8) =   8.500
Qbr(  9) =    .000
Qbr( 10) =   8.500
Qbr( 11) =    .000
Qbr( 12) =    .000
Qbr( 13) =    .000
Qbr( 14) =  15.100
Qbr( 15) =  15.100
Qbr( 16) =  15.100
Qbr( 17) =  30.200
Qbr( 18) =    .000
Qbr( 19) =    .000
```

Qbr(20) = .000
Qbr(21) = .000
Qbr(22) = .000
Qbr(23) = .000
Qbr(24) = .000
Qbr(25) = .000
Qbr(26) = 15.100
Qbr(27) = .000
Qbr(28) = .000
Qbr(29) = .000
Qbr(30) = .000
Qbr(31) = 8.500
Qbr(32) = 8.500
Qbr(33) = 15.100
Qbr(34) = 23.600
Qbr(35) = 30.200
Qbr(36) = 23.600
Qbr(37) = 23.600
Qbr(38) = 23.600
Qbr(39) = .000
Qbr(40) = .000
Qbr(41) = .000
Qbr(42) = .000
Qbr(43) = .000
Qbr(44) = .000
Qbr(45) = 8.500
Qbr(46) = 8.500
Qbr(47) = 8.500
Qbr(48) = 8.500
Qbr(49) = .000
Qbr(50) = 8.500
Qbr(51) = 8.500
Qbr(52) = 8.500
Qbr(53) = 53.800
Qbr(54) = 8.500
Qbr(55) = .000
Qbr(56) = 15.100
Qbr(57) = 62.300
Qbr(58) = 15.100
Qbr(59) = 8.500
Qbr(60) = 8.500
Qbr(61) = 8.500
Qbr(62) = 8.500
Qbr(63) = 8.500
Qbr(64) = 8.500
Qbr(65) = 8.500
Qbr(66) = 8.500
Qbr(67) = 8.500
Qbr(68) = 8.500
Qbr(69) = 8.500
Qbr(70) = 28.600
Qbr(71) = 85.900
Qbr(72) = 28.600
Qbr(73) = .000
Qbr(74) = .000
Qbr(75) = .000
Qbr(76) = 20.100
Qbr(77) = .000
Qbr(78) = .000
Qbr(79) = 114.500
Qbr(80) = .000
Qbr(81) = 15.100
Qbr(82) = 114.500
Qbr(83) = 15.100
Qbr(84) = 15.100
Qbr(85) = 144.700
Qbr(86) = 144.700
Qbr(87) = 144.700
Qbr(88) = 10.100
Qbr(89) = 10.100
Qbr(90) = 10.100
Qbr(91) = 10.100
Qbr(92) = 144.700
Qbr(93) = 10.100
Qbr(94) = 154.800
Qbr(95) = .000
Qbr(96) = .000
Qbr(97) = 20.100
Qbr(98) = 20.100
Qbr(99) = 20.100
Qbr(100) = 154.800
Qbr(101) = .000
Qbr(102) = .000
Qbr(103) = 20.100
Qbr(104) = 20.100
Qbr(105) = 154.800
Qbr(106) = 20.100
Qbr(107) = .000

Qbr(108) = 174.900
Qbr(109) = .000
Qbr(110) = 174.900

Network temperatures (C)

Tbr(1) = 35.000
Tbr(2) = 35.000
Tbr(3) = 35.000
Tbr(4) = 35.000
Tbr(5) = 35.000
Tbr(6) = 35.000
Tbr(7) = 35.000
Tbr(8) = 30.590
Tbr(9) = .000
Tbr(10) = 30.590
Tbr(11) = .000
Tbr(12) = 20.000
Tbr(13) = 20.000
Tbr(14) = 35.000
Tbr(15) = 35.000
Tbr(16) = 35.000
Tbr(17) = 35.000
Tbr(18) = 20.000
Tbr(19) = 20.000
Tbr(20) = 20.000
Tbr(21) = 20.000
Tbr(22) = 20.000
Tbr(23) = 20.000
Tbr(24) = 20.000
Tbr(25) = .000
Tbr(26) = 35.000
Tbr(27) = 20.000
Tbr(28) = 20.000
Tbr(29) = 20.000
Tbr(30) = .000
Tbr(31) = 30.590
Tbr(32) = 30.590
Tbr(33) = 35.000
Tbr(34) = 33.412
Tbr(35) = 35.000
Tbr(36) = 33.412
Tbr(37) = 33.412
Tbr(38) = 33.412
Tbr(39) = 20.000
Tbr(40) = 20.000
Tbr(41) = 20.000
Tbr(42) = 20.000
Tbr(43) = .000
Tbr(44) = 20.000
Tbr(45) = 30.590
Tbr(46) = 30.590
Tbr(47) = 30.590
Tbr(48) = 30.590
Tbr(49) = 20.000
Tbr(50) = 30.590
Tbr(51) = 30.590
Tbr(52) = 30.590
Tbr(53) = 34.303
Tbr(54) = 30.590
Tbr(55) = .000
Tbr(56) = 35.000
Tbr(57) = 33.797
Tbr(58) = 35.000
Tbr(59) = 30.590
Tbr(60) = 30.590
Tbr(61) = 30.590
Tbr(62) = 30.590
Tbr(63) = 30.590
Tbr(64) = 30.590
Tbr(65) = 30.590
Tbr(66) = 30.590
Tbr(67) = 30.590
Tbr(68) = 30.590
Tbr(69) = 30.590
Tbr(70) = 33.689
Tbr(71) = 33.691
Tbr(72) = 33.689
Tbr(73) = 20.000
Tbr(74) = .000
Tbr(75) = .000
Tbr(76) = 35.000
Tbr(77) = .000
Tbr(78) = 20.000
Tbr(79) = 33.690
Tbr(80) = 20.000
Tbr(81) = 35.000
Tbr(82) = 33.690

Tbr(83) = 35.000
Tbr(84) = 35.000
Tbr(85) = 33.964
Tbr(86) = 33.964
Tbr(87) = 33.964
Tbr(88) = 35.000
Tbr(89) = 35.000
Tbr(90) = 35.000
Tbr(91) = 35.000
Tbr(92) = 33.964
Tbr(93) = 35.000
Tbr(94) = 34.031
Tbr(95) = .000
Tbr(96) = .000
Tbr(97) = 35.000
Tbr(98) = 35.000
Tbr(99) = 35.000
Tbr(100) = 34.031
Tbr(101) = 20.000
Tbr(102) = .000
Tbr(103) = 35.000
Tbr(104) = 35.000
Tbr(105) = 34.031
Tbr(106) = 35.000
Tbr(107) = .000
Tbr(108) = 34.143
Tbr(109) = .000
Tbr(110) = 34.143

Network air flows (L/s)

Qgas(1)= .556
Qgas(2)= .556
Qgas(3)= .556
Qgas(4)= .556
Qgas(5)= .556
Qgas(6)= .556
Qgas(7)= .556
Qgas(8)= .556
Qgas(9)= .556
Qgas(10)= .556
Qgas(11)= .556
Qgas(12)= .000
Qgas(13)= .556
Qgas(14)= .556
Qgas(15)= .000
Qgas(16)= .000
Qgas(17)= .556
Qgas(18)= .556
Qgas(19)= .556
Qgas(20)= .556
Qgas(21)= .556
Qgas(22)= .556
Qgas(23)= .556
Qgas(24)= .556
Qgas(25)= .556
Qgas(26)= .556
Qgas(27)= .000
Qgas(28)= .556
Qgas(29)= .556
Qgas(30)= .556
Qgas(31)= .000
Qgas(32)= 1.970
Qgas(33)= .556
Qgas(34)= .000
Qgas(35)= .556
Qgas(36)= .556
Qgas(37)= .556
Qgas(38)= .556
Qgas(39)= .556
Qgas(40)= .556
Qgas(41)= .556
Qgas(42)= .000
Qgas(43)= 1.970
Qgas(44)= 1.970
Qgas(45)= 1.970
Qgas(46)= 1.970
Qgas(47)= 1.970
Qgas(48)= 1.970
Qgas(49)= 1.970
Qgas(50)= .556
Qgas(51)= .556
Qgas(52)= 1.970
Qgas(53)= .556
Qgas(54)= 1.970
Qgas(55)= 1.970
Qgas(56)= 1.970
Qgas(57)= 1.970

```
Qgas( 58)=     .000
Qgas( 59)=     .000
Qgas( 60)=    1.970
Qgas( 61)=    1.970
Qgas( 62)=    1.970
Qgas( 63)=    1.970
Qgas( 64)=    1.970
Qgas( 65)=     .000
Qgas( 66)=    1.970
Qgas( 67)=    1.970
Qgas( 68)=    1.970
Qgas( 69)=    1.970
Qgas( 70)=    1.970
Qgas( 71)=    1.970
Qgas( 72)=     .000
Qgas( 73)=     .000
Qgas( 74)=     .000
Qgas( 75)=    1.970
Qgas( 76)=    1.970
Qgas( 77)=     .000
Qgas( 78)=    1.970
Qgas( 79)=    1.970
Qgas( 80)=    1.970
Qgas( 81)=     .000
Qgas( 82)=    1.970
Qgas( 83)=    1.970

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Emissions calculations : Equilibrium assumption
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Network concentrations (mg/L)

Cbr(  1) =    .0000
Cbr(  2) =    .0000
Cbr(  3) =    .0000
Cbr(  4) =    .0000
Cbr(  5) =    .0000
Cbr(  6) =    .0000
Cbr(  7) =    .0000
Cbr(  8) =    .3344
Cbr(  9) =    .0000
Cbr( 10) =    .3285
Cbr( 11) =    .0000
Cbr( 12) =    .0000
Cbr( 13) =    .0000
Cbr( 14) =    .0000
Cbr( 15) =    .0000
Cbr( 16) =    .0000
Cbr( 17) =    .0000
Cbr( 18) =    .0000
Cbr( 19) =    .0000
Cbr( 20) =    .0000
Cbr( 21) =    .0000
Cbr( 22) =    .0000
Cbr( 23) =    .0000
Cbr( 24) =    .0000
Cbr( 25) =    .0000
Cbr( 26) =    .0000
Cbr( 27) =    .0000
Cbr( 28) =    .0000
Cbr( 29) =    .0000
Cbr( 30) =    .0000
Cbr( 31) =    .3227
Cbr( 32) =    .3227
Cbr( 33) =    .0000
Cbr( 34) =    .1220
Cbr( 35) =    .0000
Cbr( 36) =    .1174
Cbr( 37) =    .1212
Cbr( 38) =    .1203
Cbr( 39) =    .0000
Cbr( 40) =    .0000
Cbr( 41) =    .0000
Cbr( 42) =    .0000
Cbr( 43) =    .0000
Cbr( 44) =    .0000
Cbr( 45) =    .3171
Cbr( 46) =    .3412
Cbr( 47) =    .3473
Cbr( 48) =    .3115
Cbr( 49) =    .0000
Cbr( 50) =    .3060
Cbr( 51) =    .3060
Cbr( 52) =    .2877
Cbr( 53) =    .0515
Cbr( 54) =    .2705
Cbr( 55) =    .0000
Cbr( 56) =    .0000
```

Cbr(57) = .0806
Cbr(58) = .0000
Cbr(59) = .4364
Cbr(60) = .4641
Cbr(61) = .5343
Cbr(62) = .5249
Cbr(63) = .4936
Cbr(64) = .5439
Cbr(65) = .5537
Cbr(66) = .5870
Cbr(67) = .5520
Cbr(68) = .5190
Cbr(69) = .4881
Cbr(70) = .1421
Cbr(71) = .1010
Cbr(72) = .1421
Cbr(73) = .0000
Cbr(74) = .0000
Cbr(75) = .0000
Cbr(76) = .0000
Cbr(77) = .0000
Cbr(78) = .0000
Cbr(79) = .1112
Cbr(80) = .0000
Cbr(81) = .0000
Cbr(82) = .1107
Cbr(83) = .0000
Cbr(84) = .0000
Cbr(85) = .0876
Cbr(86) = .0872
Cbr(87) = .0868
Cbr(88) = .0000
Cbr(89) = .0000
Cbr(90) = .0000
Cbr(91) = .0000
Cbr(92) = .0865
Cbr(93) = .0000
Cbr(94) = .0808
Cbr(95) = .0000
Cbr(96) = .0000
Cbr(97) = .0000
Cbr(98) = .0000
Cbr(99) = .0000
Cbr(100) = .0808
Cbr(101) = .0000
Cbr(102) = .0000
Cbr(103) = .0000
Cbr(104) = .0000
Cbr(105) = .0805
Cbr(106) = .0000
Cbr(107) = .0000
Cbr(108) = .0710
Cbr(109) = .0000
Cbr(110) = .0708

Mass emissions from node (mg/s) 1= .000
Mass emissions from node (mg/s) 2= .000
Mass emissions from node (mg/s) 3= .000
Mass emissions from node (mg/s) 4= .000
Mass emissions from node (mg/s) 5= .000
Mass emissions from node (mg/s) 6= .000
Mass emissions from node (mg/s) 7= .000
Mass emissions from node (mg/s) 8= .050
Mass emissions from node (mg/s) 9= .049
Mass emissions from node (mg/s) 10= .000
Mass emissions from node (mg/s) 11= .000
Mass emissions from node (mg/s) 12= .000
Mass emissions from node (mg/s) 13= .000
Mass emissions from node (mg/s) 14= .000
Mass emissions from node (mg/s) 15= .000
Mass emissions from node (mg/s) 16= .000
Mass emissions from node (mg/s) 17= .000
Mass emissions from node (mg/s) 18= .000
Mass emissions from node (mg/s) 19= .000
Mass emissions from node (mg/s) 20= .000
Mass emissions from node (mg/s) 21= .000
Mass emissions from node (mg/s) 22= .000
Mass emissions from node (mg/s) 23= .000
Mass emissions from node (mg/s) 24= .000
Mass emissions from node (mg/s) 25= .000
Mass emissions from node (mg/s) 26= .000
Mass emissions from node (mg/s) 27= .000
Mass emissions from node (mg/s) 28= .048
Mass emissions from node (mg/s) 29= .020
Mass emissions from node (mg/s) 30= .020
Mass emissions from node (mg/s) 31= .000
Mass emissions from node (mg/s) 32= .069
Mass emissions from node (mg/s) 33= .020

Mass emissions from node (mg/s) 34= .000
Mass emissions from node (mg/s) 35= .000
Mass emissions from node (mg/s) 36= .000
Mass emissions from node (mg/s) 37= .000
Mass emissions from node (mg/s) 38= .000
Mass emissions from node (mg/s) 39= .047
Mass emissions from node (mg/s) 40= .052
Mass emissions from node (mg/s) 41= .047
Mass emissions from node (mg/s) 42= .000
Mass emissions from node (mg/s) 43= .155
Mass emissions from node (mg/s) 44= .146
Mass emissions from node (mg/s) 45= .048
Mass emissions from node (mg/s) 46= .000
Mass emissions from node (mg/s) 47= .060
Mass emissions from node (mg/s) 48= .235
Mass emissions from node (mg/s) 49= .250
Mass emissions from node (mg/s) 50= .080
Mass emissions from node (mg/s) 51= .081
Mass emissions from node (mg/s) 52= .266
Mass emissions from node (mg/s) 53= .083
Mass emissions from node (mg/s) 54= .298
Mass emissions from node (mg/s) 55= .280
Mass emissions from node (mg/s) 56= .263
Mass emissions from node (mg/s) 57= .084
Mass emissions from node (mg/s) 58= .000
Mass emissions from node (mg/s) 59= .000
Mass emissions from node (mg/s) 60= .000
Mass emissions from node (mg/s) 61= .000
Mass emissions from node (mg/s) 62= .000
Mass emissions from node (mg/s) 63= .066
Mass emissions from node (mg/s) 64= .000
Mass emissions from node (mg/s) 65= .000
Mass emissions from node (mg/s) 66= .052
Mass emissions from node (mg/s) 67= .052
Mass emissions from node (mg/s) 68= .052
Mass emissions from node (mg/s) 69= .000
Mass emissions from node (mg/s) 70= .000
Mass emissions from node (mg/s) 71= .000
Mass emissions from node (mg/s) 72= .000
Mass emissions from node (mg/s) 73= .000
Mass emissions from node (mg/s) 74= .000
Mass emissions from node (mg/s) 75= .000
Mass emissions from node (mg/s) 76= .000
Mass emissions from node (mg/s) 77= .000
Mass emissions from node (mg/s) 78= .048
Mass emissions from node (mg/s) 79= .000
Mass emissions from node (mg/s) 80= .000
Mass emissions from node (mg/s) 81= .000
Mass emissions from node (mg/s) 82= .043
Mass emissions from node (mg/s) 83= .043

Network totals

Total mass input to system (mg/s)= 15.491

Total network emissions (mg/s)= 3.111

Stripping Efficiency (%) = 20.08

Example naUTilus OSBL output file
(ondout.txt)

```
Node,Emissions,Gas_flow
 1, .0000000000, .556
 2, .0000000000, .556
 3, .0000000000, .556
 4, .0000000000, .556
 5, .0000000000, .556
 6, .0000000000, .556
 7, .0000000000, .556
 8, .0500768425, .556
 9, .0491946513, .556
10, .0000000000, .556
11, .0000000000, .556
12, .0000000000, .000
13, .0000000000, .556
14, .0000000000, .556
15, .0000000000, .000
16, .0000000000, .000
17, .0000000000, .556
18, .0000000000, .556
19, .0000000000, .556
20, .0000000000, .556
21, .0000000000, .556
22, .0000000000, .556
23, .0000000000, .556
24, .0000000000, .556
25, .0000000000, .556
26, .0000000000, .556
27, .0000000000, .000
28, .0483280015, .556
29, .0203054677, .556
30, .0201633097, .556
31, .0000000000, .000
32, .0691516058, 1.970
33, .0200221469, .556
34, .0000000000, .000
35, .0000000000, .556
36, .0000000000, .556
37, .0000000000, .556
38, .0000000000, .556
39, .0474766192, .556
40, .0520115139, .556
41, .0466402355, .556
42, .0000000000, .000
43, .1552489631, 1.970
44, .1459819132, 1.970
45, .0480474180, 1.970
46, .0000000000, 1.970
47, .0599749395, 1.970
48, .2354908780, 1.970
49, .2504400293, 1.970
50, .0800138983, .556
51, .0814487607, .556
52, .2663381649, 1.970
53, .0829093539, .556
54, .2978512405, 1.970
55, .2800720408, 1.970
56, .2633541090, 1.970
57, .0844166043, 1.970
58, .0000000000, .000
59, .0000000000, .000
60, .0000000000, 1.970
61, .0000000000, 1.970
62, .0000000000, 1.970
63, .0657391519, 1.970
64, .0000000000, 1.970
65, .0000000000, .000
66, .0522404484, 1.970
67, .0520250591, 1.970
68, .0518105579, 1.970
69, .0000000000, 1.970
70, .0000000000, 1.970
71, .0000000000, 1.970
72, .0000000000, .000
73, .0000000000, .000
74, .0000000000, .000
75, .0000000000, 1.970
76, .0000000000, 1.970
77, .0000000000, .000
78, .0483430805, 1.970
79, .0000000000, 1.970
80, .0000000000, 1.970
81, .0000000000, .000
82, .0427860562, 1.970
83, .0426392046, 1.970
```

Example naUTilus OSBL output file
(obrout.txt)

```
Branch,Liq_flow,Temp,Gas_flow,Liq_conc,Kla
  1,  15.100,   35.00,   .5564,  .00000, .000044
  2,  15.100,   35.00,   .5564,  .00000, .000043
  3,  15.100,   35.00,   .5564,  .00000, .000043
  4,  15.100,   35.00,   .5564,  .00000, .000043
  5,  15.100,   35.00,   .5564,  .00000, .000043
  6,  15.100,   35.00,   .5564,  .00000, .000043
  7,  15.100,   35.00,   .5564,  .00000, .000043
  8,   8.500,   30.59,   .2782,  .33442, .000037
  9,    .000,     .00,   .2782,  .00000, .000000
 10,   8.500,   30.59,   .2782,  .32853, .000037
 11,    .000,     .00,   .5564,  .00000, .000000
 12,    .000,   20.00,   .5564,  .00000, .000000
 13,    .000,   20.00,   .0000,  .00000, .000000
 14,  15.100,   35.00,   .2782,  .00000, .000043
 15,  15.100,   35.00,   .2782,  .00000, .000044
 16,  15.100,   35.00,   .5564,  .00000, .000044
 17,  30.200,   35.00,   .0000,  .00000, .000048
 18,    .000,   20.00,   .0000,  .00000, .000000
 19,    .000,   20.00,   .5564,  .00000, .000000
 20,    .000,   20.00,   .0000,  .00000, .000000
 21,    .000,   20.00,   .5564,  .00000, .000000
 22,    .000,   20.00,   .5564,  .00000, .000000
 23,    .000,   20.00,   .5564,  .00000, .000000
 24,    .000,   20.00,   .5564,  .00000, .000000
 25,    .000,     .00,   .5564,  .00000, .000000
 26,  15.100,   35.00,   .5564,  .00000, .000044
 27,    .000,   20.00,   .5564,  .00000, .000000
 28,    .000,   20.00,   .5564,  .00000, .000000
 29,    .000,   20.00,   .2782,  .00000, .000000
 30,    .000,     .00,   .5564,  .00000, .000000
 31,   8.500,   30.59,   .0000,  .32274, .000037
 32,   8.500,   30.59,   .5564,  .32274, .000037
 33,  15.100,   35.00,   .2782,  .00000, .000044
 34,  23.600,   33.41,   .5564,  .12204, .000046
 35,  30.200,   35.00,   .0000,  .00000, .000048
 36,  23.600,   33.41,   .0000,  .11740, .000046
 37,  23.600,   33.41,   .5564,  .12118, .000046
 38,  23.600,   33.41,  1.9698,  .12033, .000046
 39,    .000,   20.00,   .0000,  .00000, .000000
 40,    .000,   20.00,   .0000,  .00000, .000000
 41,    .000,   20.00,   .5564,  .00000, .000000
 42,    .000,   20.00,   .5564,  .00000, .000000
 43,    .000,     .00,   .5564,  .00000, .000000
 44,    .000,   20.00,   .5564,  .00000, .000000
 45,   8.500,   30.59,   .5564,  .31706, .000037
 46,   8.500,   30.59,   .2782,  .34122, .000037
 47,   8.500,   30.59,   .5564,  .34734, .000037
 48,   8.500,   30.59,   .5564,  .31147, .000037
 49,    .000,   20.00,   .0000,  .00000, .000000
 50,   8.500,   30.59,   .0000,  .30598, .000035
 51,   8.500,   30.59,  1.9698,  .30598, .000035
 52,   8.500,   30.59,  1.9698,  .28772, .000035
 53,  53.800,   34.30,   .6566,  .05150, .000051
 54,   8.500,   30.59,   .6566,  .27054, .000035
 55,    .000,     .00,   .6566,  .00000, .000000
 56,  15.100,   35.00,  1.9698,  .00000, .000044
 57,  62.300,   33.80,   .6566,  .08061, .000052
 58,  15.100,   35.00,   .6566,  .00000, .000044
 59,   8.500,   30.59,   .6566,  .43643, .000035
 60,   8.500,   30.59,  1.9698,  .46413, .000035
 61,   8.500,   30.59,   .5564,  .53434, .000035
 62,   8.500,   30.59,  1.9698,  .52493, .000035
 63,   8.500,   30.59,  1.9698,  .49360, .000035
 64,   8.500,   30.59,   .5564,  .54393, .000035
 65,   8.500,   30.59,   .5564,  .55368, .000037
 66,   8.500,   30.59,  1.9698,  .58704, .000035
 67,   8.500,   30.59,  1.9698,  .55200, .000035
 68,   8.500,   30.59,  1.9698,  .51905, .000035
 69,   8.500,   30.59,   .9849,  .48807, .000035
 70,  28.600,   33.69,   .0000,  .14210, .000046
 71,  85.900,   33.69,   .0000,  .10095, .000054
 72,  28.600,   33.69,   .0000,  .14210, .000046
 73,    .000,   20.00,   .0000,  .00000, .000000
 74,    .000,     .00,  1.9698,  .00000, .000000
 75,    .000,     .00,   .0000,  .00000, .000000
 76,  20.100,   35.00,   .9849,  .00000, .000046
 77,    .000,     .00,  1.9698,  .00000, .000000
 78,    .000,   20.00,  1.9698,  .00000, .000000
 79, 114.500,   33.69,   .9849,  .11123, .000057
 80,    .000,   20.00,   .9849,  .00000, .000000
 81,  15.100,   35.00,  1.9698,  .00000, .000044
 82, 114.500,   33.69,   .0000,  .11066, .000057
 83,  15.100,   35.00,   .0000,  .00000, .000044
 84,  15.100,   35.00,   .0000,  .00000, .000044
```

85,	144.700,	33.96,	1.9698,	.08756,	.000059
86,	144.700,	33.96,	1.9698,	.08720,	.000059
87,	144.700,	33.96,	1.9698,	.08684,	.000059
88,	10.100,	35.00,	1.9698,	.00000,	.000042
89,	10.100,	35.00,	1.9698,	.00000,	.000042
90,	10.100,	35.00,	1.9698,	.00000,	.000042
91,	10.100,	35.00,	.0000,	.00000,	.000042
92,	144.700,	33.96,	.0000,	.08648,	.000059
93,	10.100,	35.00,	.0000,	.00000,	.000042
94,	154.800,	34.03,	.0000,	.08084,	.000060
95,	.000,	.00,	.0000,	.00000,	.000000
96,	.000,	.00,	.0000,	.00000,	.000000
97,	20.100,	35.00,	1.9698,	.00000,	.000046
98,	20.100,	35.00,	1.9698,	.00000,	.000046
99,	20.100,	35.00,	.0000,	.00000,	.000046
100,	154.800,	34.03,	.9849,	.08084,	.000060
101,	.000,	20.00,	.9849,	.00000,	.000000
102,	.000,	.00,	1.9698,	.00000,	.000000
103,	20.100,	35.00,	1.9698,	.00000,	.000046
104,	20.100,	35.00,	.0000,	.00000,	.000046
105,	154.800,	34.03,	.6566,	.08053,	.000060
106,	20.100,	35.00,	.6566,	.00000,	.000046
107,	.000,	.00,	.6566,	.00000,	.000000
108,	174.900,	34.14,	.9849,	.07103,	.000061
109,	.000,	.00,	.9849,	.00000,	.000000
110,	174.900,	34.14,	.0000,	.07079,	.000061

Appendix E

USER DOCUMENTATION AND EXERCISE

User Document
Center for Research in Water Resources &
Center for Energy Studies - UT Program in Air Resources Engineering
University of Texas at Austin

Using ArcView as a User Interface for naUTilus

By Cindy How

c.how@mail.utexas.edu

Table of Contents

Introduction

This document comes in two parts: a general procedure document and a user exercise. The general procedure document discusses basic file set up and the use of the *naUTilus*/ArcView interface. The user exercise runs through a specific example applying the interface. This document does not contain the technical documentation for the naUTilus model, which is available separately.

The *naUTilus* model

The model was developed as a tool to predict fugitive emissions of volatile organic compounds (VOCs) from industrial sewer networks. It consists of two FORTRAN modules. One module estimates emissions from the point of discharge in specific process units, termed "inside the battery limit" (ISBL). The other module estimates emissions in the main collection system, termed "outside the battery limit" (OSBL). For further discussion of the naUTilus model, see the Technical Documentation (Olson, *et al.*, 1997).

The *naUTilus*/ArcView interface

The *naUTilus*/ArcView interface was developed to facilitate the application of *naUTilus* to large sewer networks. It is enabled by a series of programs (scripts) written in ArcView's object-oriented programming language, Avenue. Data input for the *naUTilus* model, execution of *naUTilus* model, and display of *naUTilus* results are done in ArcView.

Requirements

Data required for *naUTilus* includes flow characteristics, chemical properties, sewer dimensions, and ambient conditions. Also needed for the *naUTilus*/ArcView interface are digital representations of the industrial sewer network

The interface was written for ArcView version 3.0a., distributed by ESRI (Environmental Solutions Research Institute).

In addition to the basic ArcView software, the following files are necessary for the *naUTilus*/ArcView interface:

- isbl.exe - executable *naUTilus* file for ISBL systems.
- osbl.exe - executable *naUTilus* file for OSBL systems.
- nautilus.apr - ArcView project file.
- osblnd.avl - legend file for OSBL nodes
- sewerleg.avl - legend file for ISBL nodes
- sewerlg2.avl - legend file for ISBL branches
- obranch.shp, obranch.shx, obranch.dbf, onode.shp, onode.shx, onode.dbf - GIS files describing your OSBL system
- ibranch.shp, ibranch.shx, ibranch.dbf, inodes.shp, inodes.shx, inodes.dbf - GIS files describing your ISBL system. One set of the files should exist for each ISBL system. See Setting up the network files for further explanation.

The *naUTilus*/ArcView interface was set up to run for the PC version of ArcView.

Setting up the network files

For the *naUTilus*/ArcView interface to run properly, your ISBL and OSBL must be stored in a specific manner. Create a main directory in which all of the files relating to a specific OSBL site are stored. Under this main directory, create a subdirectory for each ISBL unit associated with the OSBL unit. For OSBL and ISBL units, use the names at most 8 letters in length.
The main directory should hold the following files:

- isbl.exe
- osbl.exe
- nautilus.apr
- osblnd.avl
- sewerleg.avl

- sewerlg2.avl
- obranch.shp
- obranch.dbf
- obranch.shx
- onode.shp
- onode.dbf
- onode.shx
- Subdirectories for each feeding ISBL

Each subdirectory should hold the GIS files (inodes.shp, inodes.dbf, inodes.shx, ibranch.shp, ibranch.dbf, ibranch.shx) for one ISBL system which feeds the OSBL.

Diagram A

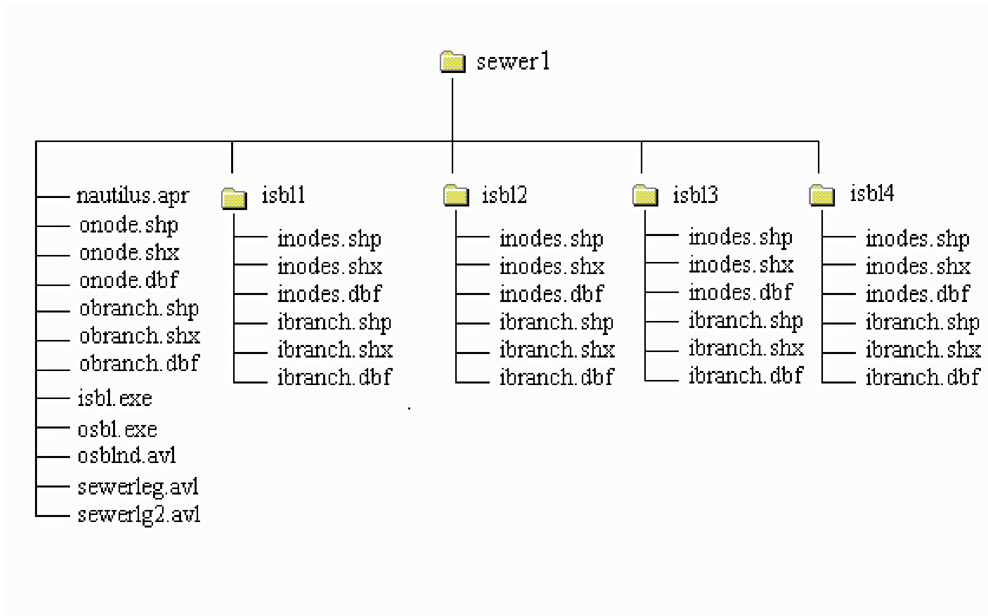


Diagram A shows the file set up for an OSBL system with ISBL feeds from plants designated "Isbl1" through "Isbl4". These files should exist before any operations are done with *naUTilus* or ArcView. Once operations begin, some new files will be created and appear in both the main directory and subdirectories.

Note that the file nautilus.apr is the original project file. An unedited copy of it should be kept at all times. As the original project file, it holds all of the programs needed to run *naUTilus* without any of the actual network files. This original, unedited copy should be kept and used for each new OSBL/ISBL sewer network. To keep nautilus.apr in its original state, use the Save As option the first time you work with it and work with the new project thereafter.

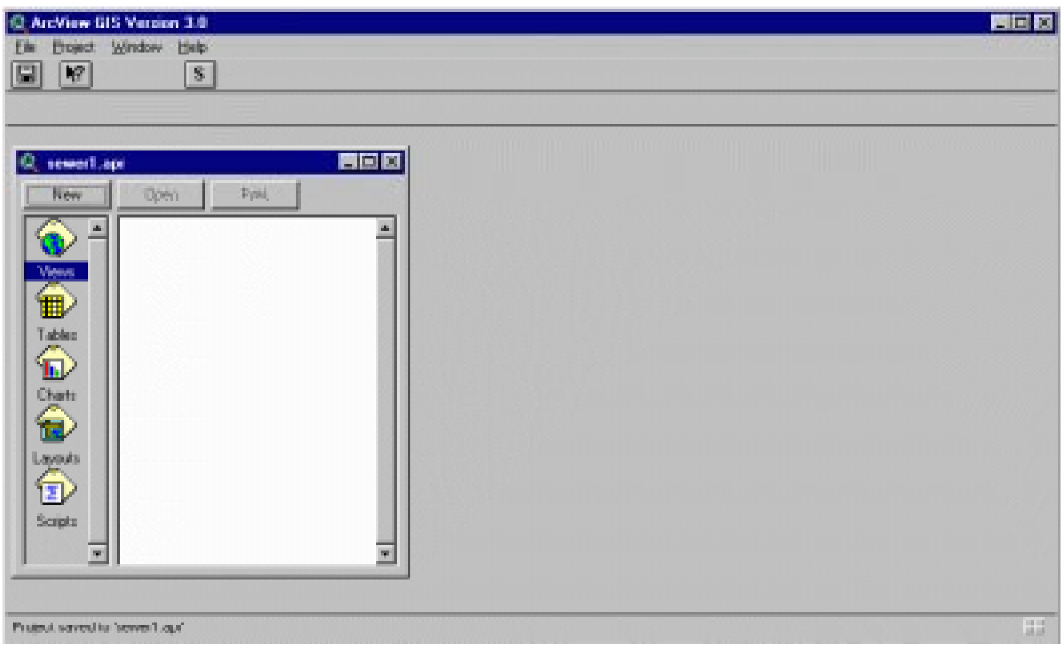
Opening the project

When first working with a sewer network in ArcView, open a project specifically for that network. To open the ArcView project, open ArcView and go to File/Open Project. Select the project **nautilus.apr**.

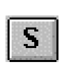
Use the Save As option, to preserve a copy of nautilus.apr in its unedited form. Go to File/Save As and name the project. Choose a name which describes your sewer network. This name should be at most 8 letters in length. The 3 letter extension to this file will be .apr (e.g. sewer1.apr).

This project file will now be the file used whenever working with that specific sewer network. The name of this project will appear in the blue bar at the top of the project window. This name will also appear as the title of the View window displaying your OSBL unit.

Note that saving the project should be done before initiating your network files



Initiating the network files

Once the ArcView project is open, bring all of the GIS data on your OSBL and ISBL units to the project by hitting the 

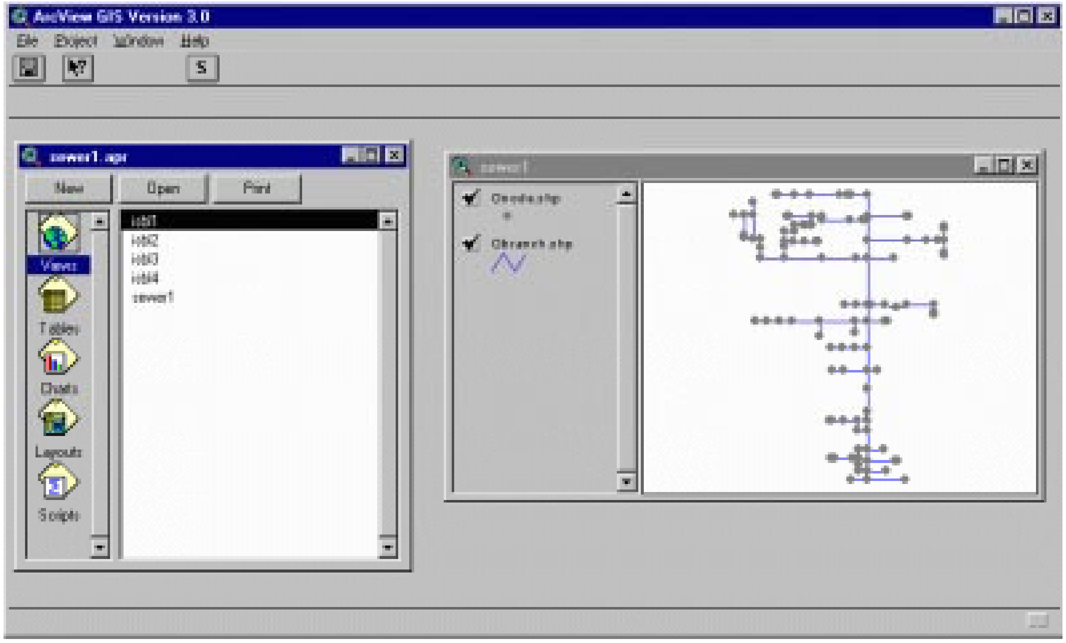
button found near the top left of the screen.

This adds the OSBL and ISBL GIS files to the project. A View window will open, displaying the map of the OSBL unit. The window name is identical to the name used when saving the project.

The reaches/branches are shown in blue and the nodes (junctions, manholes, and pipe endings) are shown in gray.

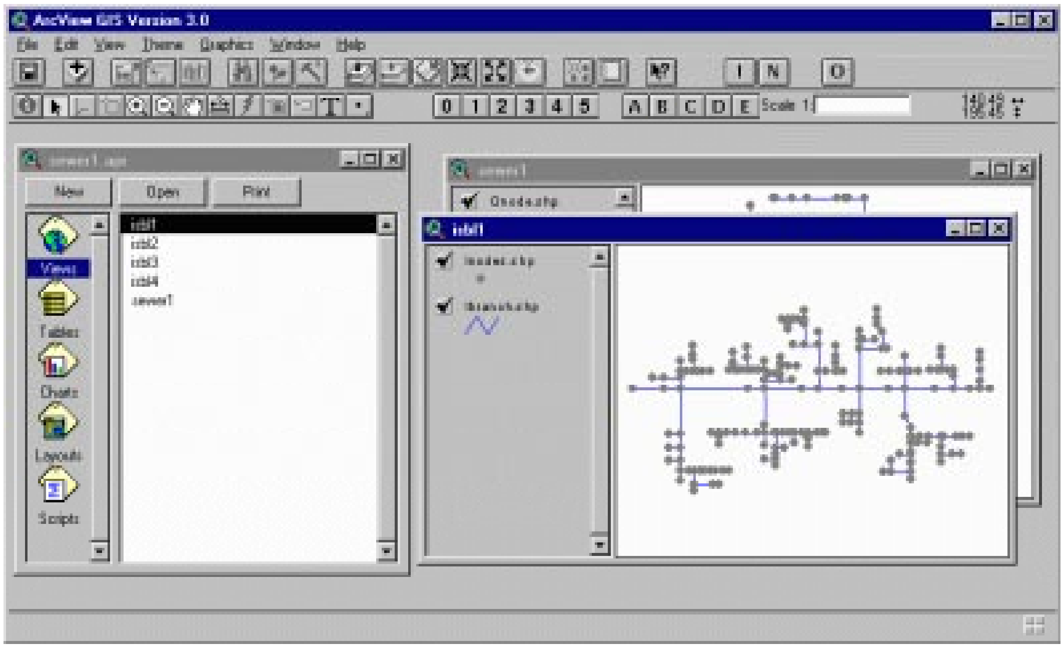
A View is added for each ISBL unit, but not opened in this step. Access the created views and through your project window by clicking on the View icon.

In addition to adding the OSBL and ISBL map data into the project, this step also creates several files in the ISBL subdirectories. These files (hardpipe.dbf, drains.dbf, drops.dbf, brtable.dbf, nodecon.dbf) should not be erased.



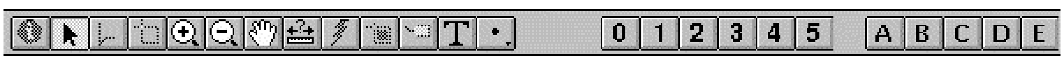
Working with ISBL

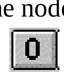
To work with any one of the ISBL units, open the View corresponding to that unit. For example, to work with Isbl1, click on Isbl1 in the project window. Click on Open. The View window for Isbl1 will open. The following figure shows the ArcView project with the OSBL view (SEWER1) and an ISBL view (ISBL1) open. Once an ISBL view is open, data can be entered for the ISBL unit.



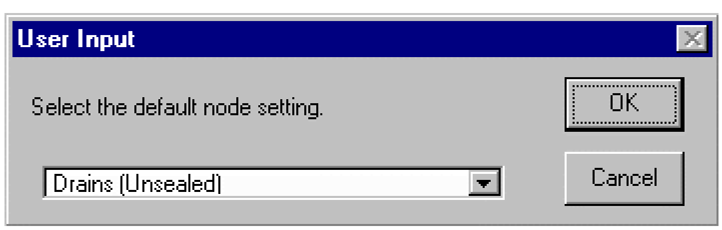
Entering ISBL data

To enter data on the ISBL network, use the buttons numbered 0 through 5 in the toolbar seen when a View window is opened. These buttons should be applied in numerical order.

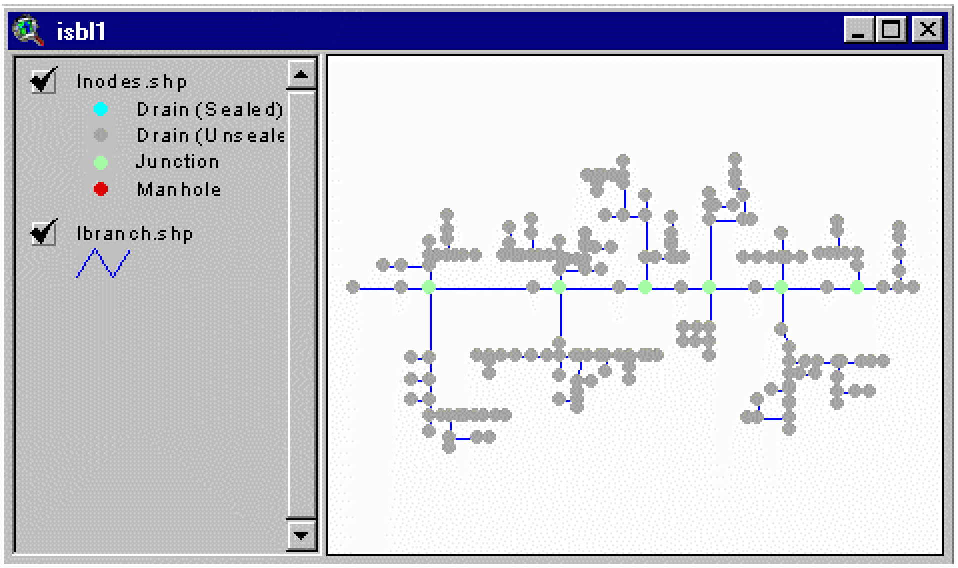


- **Step 1: Setting Drains and Junctions**
  The first step in describing an ISBL network is to select a default type for the nodes of the system. The nodes, appearing initially in gray, represent both drains and junctions. Click on the  button.

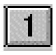
  The choices for default types are Drains (Unsealed), Junctions, or Drain/Junction (internal nodes default to junction). Disregard sealed/unsealed drain status at this step. Drains are designated as sealed or unsealed later in the process.



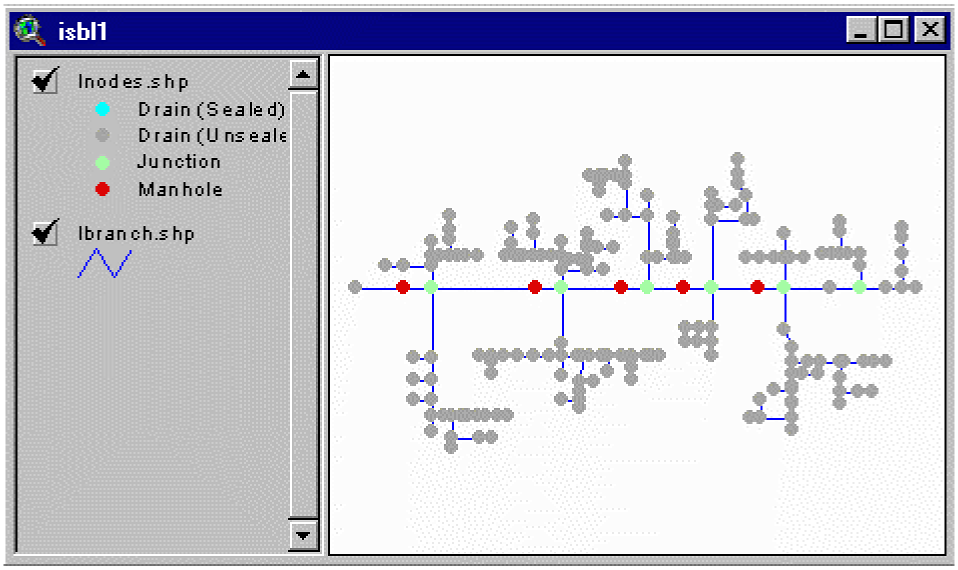
Once the default drain type is selected, edit individual drains from the default setting by clicking on the appropriate drains in the View window. Clicking again on an edited node toggles the node type back to the default setting. The legend in the left hand portion of the View indicates the colors representing each node type.



- **Step 2: Indicate Manholes**

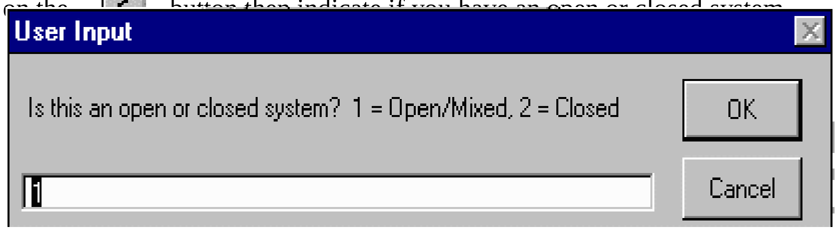
Indicate which nodes represent manholes by clicking on the  button, then clicking on the nodes that represent manholes.

Manholes are marked as indicated in the legend of the view Inodes.shp.

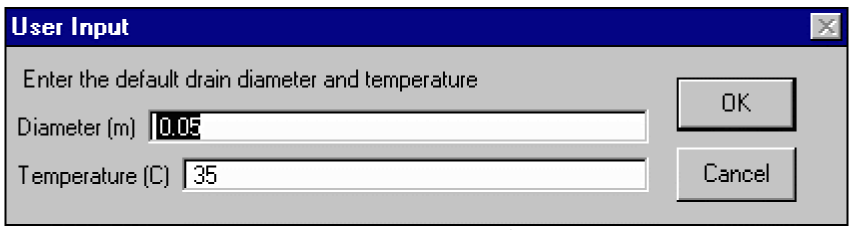


- **Step 3: Edit Drains**

Click on the button then indicate if you have an open or closed system.



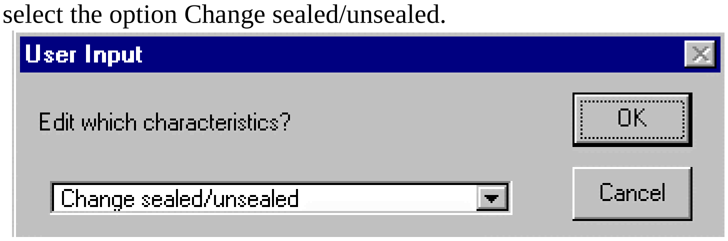
Enter a default drain diameter and temperature when prompted. These numbers will be used for all drains that are not later modified.



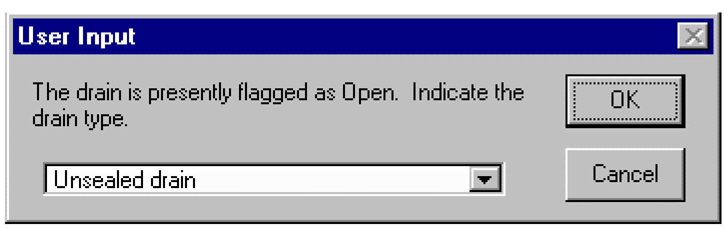
Edit the drain status (sealed or unsealed) or enter flow data at drains by selecting the drain in the View window.

- **Changing Sealed/Unsealed status**

To change a drain between an unsealed (open) drain and a sealed (closed) drain, select the option Change sealed/unsealed.



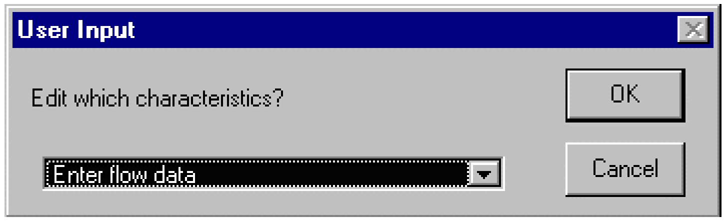
To confirm the change, ArcView presents a window indicating the present status of the drain (opened or closed).



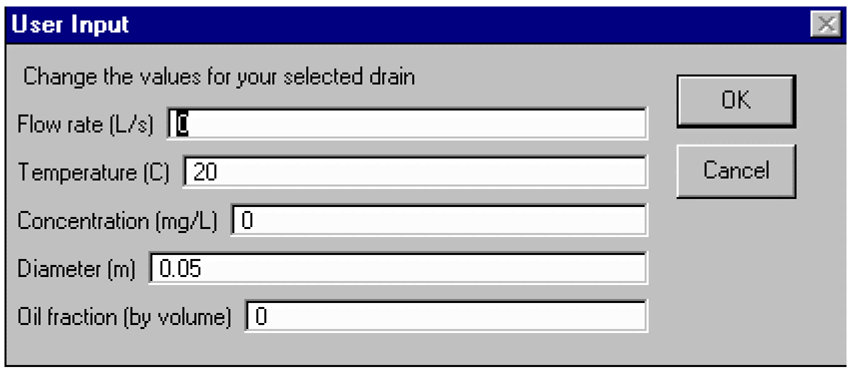
Select the correct status in this window and hit OK.

- **Entering Flow Data**

  To designate flow to a node, click on the node and select Enter flow data when prompted to choose a characteristic to edit.

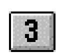


Fill in the appropriate numbers for Flow rate (L/s), Temperature (^{o}C), Concentration (mg/L), Diameter (m), and Oil fraction by volume.

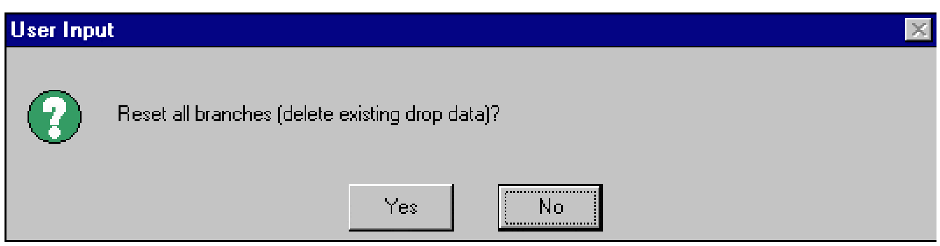


- **Step 4: Entering Drop Data**

  To enter drop data, click on the button.

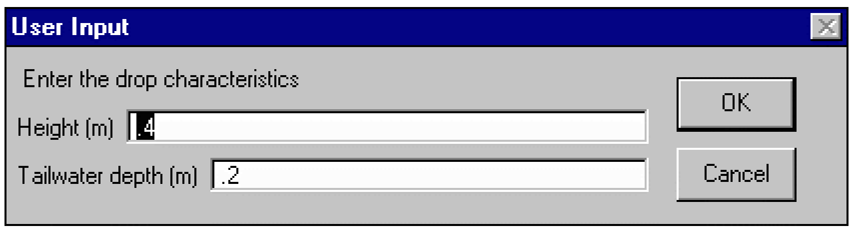


The option to reset existing drop data is presented. Choose the appropriate option. When working with a unit for first time, choose to reset the drop data.



Click on any branch at which a drop occurs when prompted. The branch associated with a drop is the branch with a drop at the terminal end. If no drops are present in the system or all drop data has been entered, move on to the next step.

To enter drop data, click on the appropriate branch. Enter data for the height of the drop and the tailwater depth. Both values should be entered in meters.

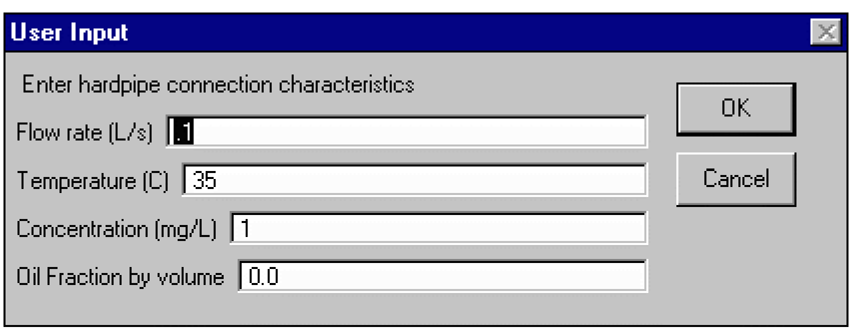


When you have completed entering the data for a given drop, the branch associated with that drop will change colors. The colors of branches with and without drops are indicated in the legend on the left hand portion of the View window.
To enter more drop data, simply click on the branch associated with that drop. Repeat these steps as many times as is necessary, then move on.

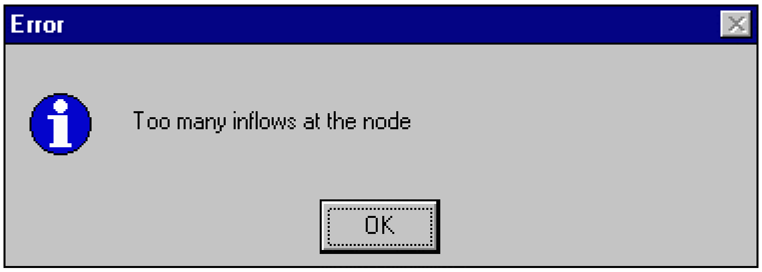
- **Step 5: Entering Hard Pipe Connections**
  To enter hard pipe connection data, click on the 

 button.

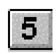
  .
  Click on a node where a hardpipe connection occurs and enter the flow rate (L/s), temperature (^{o}C), concentration (mg/L), and oil fraction of the flow through that hardpipe connection when prompted.



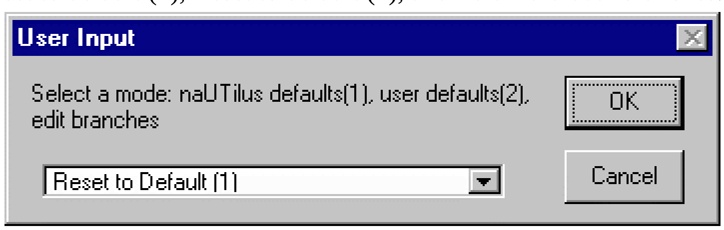
naUTilus is not configured to handle more than 4 connections at any one node or more than one hardpipe connection at a node. If you select a node at which either of these conditions occur, Arcview will give you the following error message:



- **Step 6: Editing Branches**

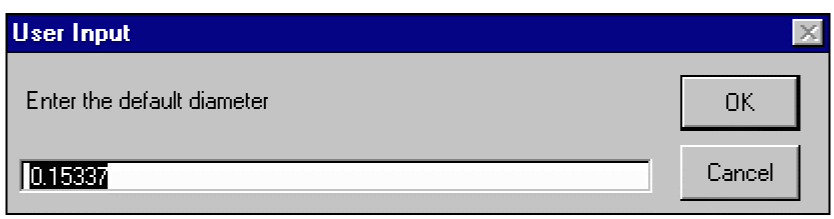
The last step in specifying ISBL characteristics is to change any branch characteristics from the default. Hit the  button and select from the three listed choices:

Reset to default (1), Reset to default (2), and Edit individual branches.



Options:

- Reset to default (1): Default (1) sets the default branch characteristics as the default specified in naUTilus. This default option sets the branch diameter of pipes flowing directly downstream from a manhole as 0.2055 meter and all other pipe diameters as 0.1534 meter. This option will reset any previously made changes to the branch default slopes or diameters.
- Reset to default (2): Default (2) sets the default branch diameter as a user chosen default. The user will be prompted to select this default. This option will reset any previously made changes to the branch default slopes or diameters. Diameters should be entered in meters.

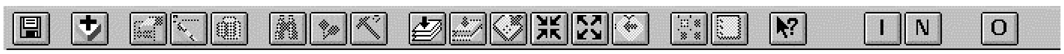


- Edit individual branches: This step is available after each of the above steps or if the user should choose to edit branch characteristics at some time after setting the defaults. It will not reset any changes made previously to branch diameters or slopes. Again, diameters should be entered in meters.

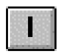
Chemical/Calculation input and Running *naUTilus* (ISBL)

After running the above steps, the user can create the ISBL *naUTilus* input file. Note that only after steps 1-6 have been completed will a complete input file be generaged. Information on the chemical being modeled, as well as which method of calculation used, is required. This information will be prompted for during creation of the *naUTilus* input file. Without this information, *naUTilus* will not run.

- **Creating the *naUTilus* input file**



To create the naUTilus input file, go to the button bar from the View window.

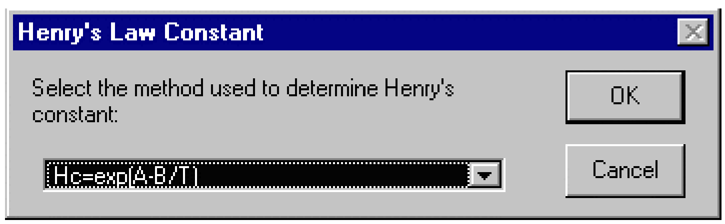
Click 

on the button in the button bar (found near the right end of the button bar). ArcView prompts for a series of data related to the inflow to your ISBL unit.

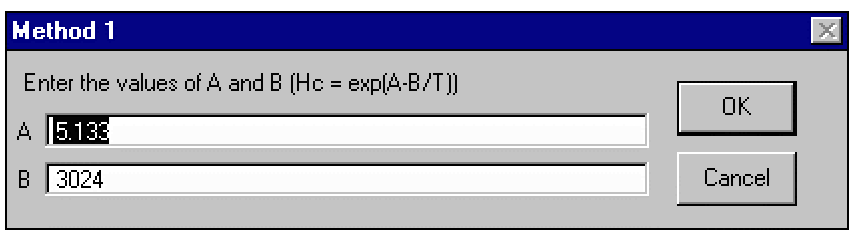
- Choosing a method to determine Henry's Law constants

  Choose one of the methods for calculating the Henry's Law constant. Provide the necessary values for this calculation.

1. Method 1: $H_c = \exp(A\text{-}B/T)$



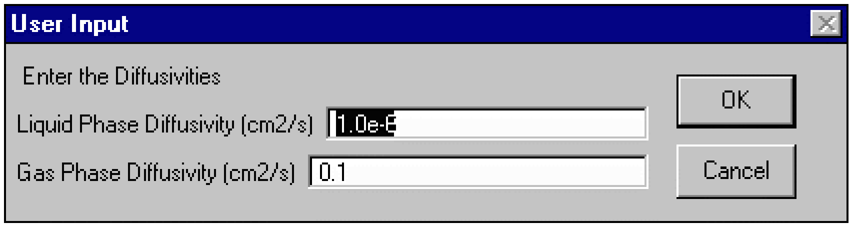
When using this method, *naUTilus* calculates the Henry's Law constant at each branch and node as a function of temperature using A and B constants. Required information: constants A and B.



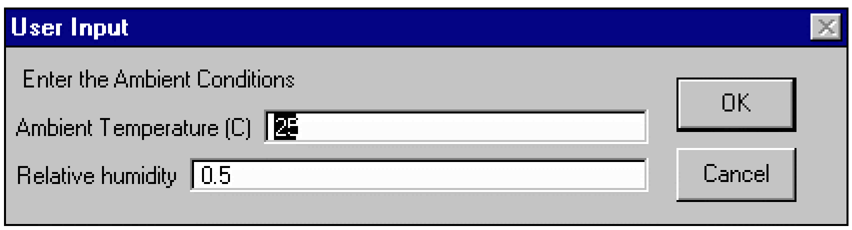
2. Method 2: H_c, Antoine constants.
   In this method, Henry's Law constant is specified at a known temperature and related to Henry's Law constant other temperatures using a ratio of vapor pressure and Antoine constants. Required information: Antoine constants (A,B,C), Henry's Law constant at a known temperature.
3. Method 3: H_c = Vp/sol using Antoine constants.
   Required information: Antoine constants (A,B,C) solubility, and molecular weight.

4. Method 4: H_c = constant.
   Required information: Henry's law constant

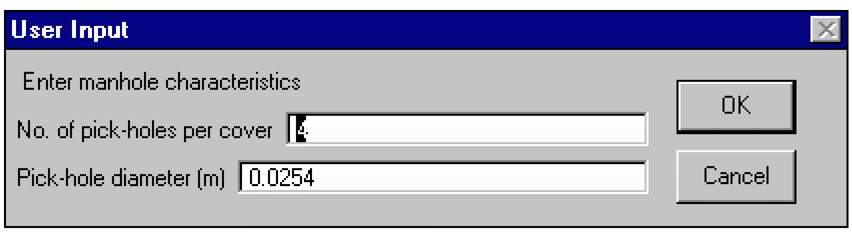
- Entering diffusivity coefficients
  Enter values for the liquid and gas phase diffusivity coefficients. These are both in cm^2/s.



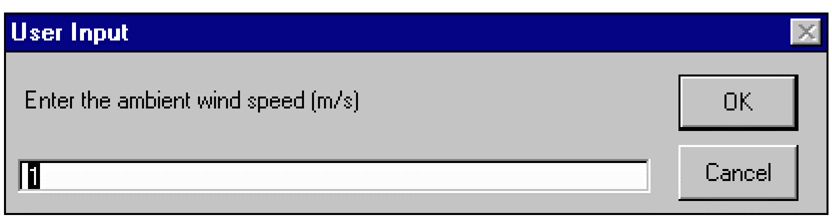
- Entering ambient temperature and humidity
  Enter the values for ambient temperature (^oC) and relative humidity. Relative humidity should be between 0 and 1 (i.e. 50% humidity is entered as 0.5). Hit OK when done.



- Enter manhole characteristics
  Input the number of pick-holes in each manhole cover as well as the pick-hole diameter in meters. Hit OK

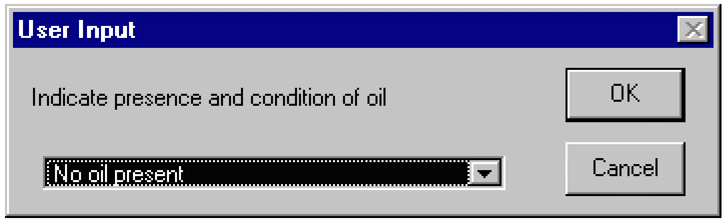


- Enter ambient wind speed
  Enter the ambient wind speed in meters per second. Hit OK.

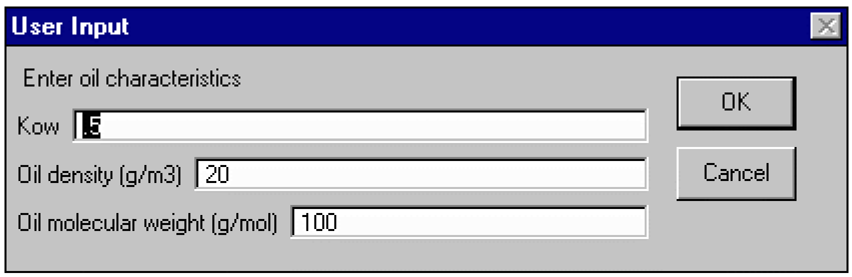


- Indicate the presence of oil
  Three conditions are defined in *naUTilus* for the presence of oil in the system.

- No oil present
- Oil present only at the surface
- Oil present and dispersed (not at the surface)

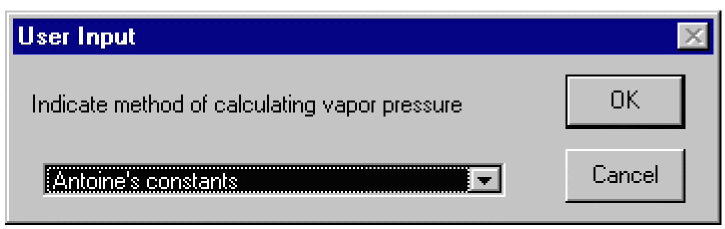


If no oil is present, ArcView will move on to the next section. For oil at the surface or dispersed in the system, ArcView will prompt for an oil-water partitioning coefficient, oil density (g/m^3), and oil molecular weight (g/mol).

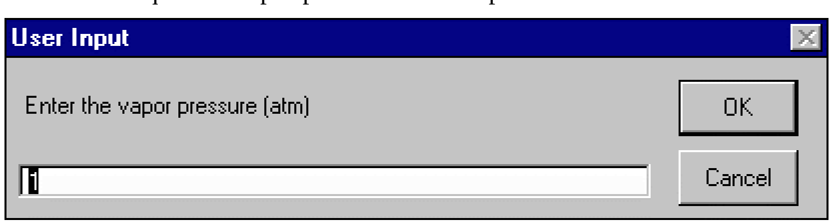


- Choosing the method to calculate vapor pressure (oil present)
  If no oil is present in the system, this step does not apply. If oil is present in the system, *naUTilus* has two options for calculating the vapor pressure: using Antoine constants or using a constant vapor pressure.

Method 1 requires that you enter the Antoine constants (A,B,C) for temperatures in Kelvin. This gives vapor pressures in mmHg.



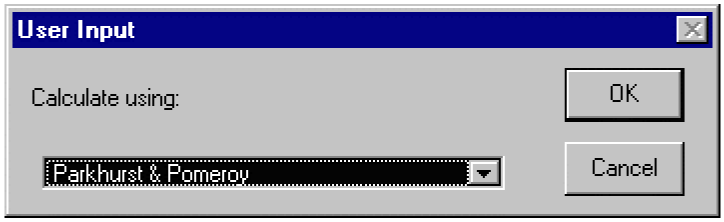
Method 2 requires a vapor pressure in atmospheres.



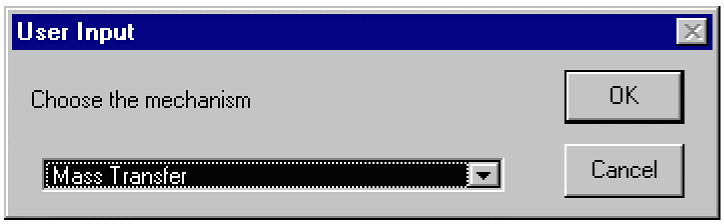
- Selecting a method to calculate reach emissions
  The two options for calculating reach emissions are:

1. Parkhurst/Pomeroy
2. Owens/Edwards/Gibbs

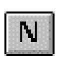
These options are discussed in the *naUTilus* user documentation (Olson *et al.*, 1997).



- Selecting the mechanism
  Indicate if you system is kinetics-limited or equilibrium-limited. See the naUTilus documentation for a discussion of the mechanisms (Olson, *et al.*, 1997)



After selecting a mechanism, all input needed for the *naUTilus* input file has been specified. ArcView writes an input file. *naUTilus* can be run on the ISBL unit.

- **Running ISBL *naUTilus***
  Once a complete input file has been created (all above steps completed in order), the *naUTilus* model can be run. To do this, have the ISBL View window active and click on the  button.

  There will be a slight delay after which *naUTilus* will run. <u>If problems running this script, occur, close other windows.</u> The problem which may arise is a shortage of memory on the device. Closing some windows or allowing some time for background processes to complete should solve this problem.

  ISBL *naUTilus* can be run for various ambient conditions and chemicals by running through the steps for executing naUTilus and entering new conditions or chemical properties

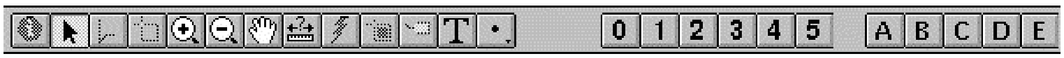
Editing ISBL data

To edit data entered for the ISBL unit, use the buttons as described for entering ISBL data. Many buttons will ask whether to reset the data. Hit "OK" only if are entering new data for every element associated with that button (i.e. resetting every drain when selecting the  button).

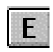
Once any data has been changed for the ISBL unit, naUTilus must be run again for the output to reflect the changes made to the ISBL unit.

Working with OSBL

The View window displaying the OSBL unit was opened initially when you initialized the network files. The name of that View window is the same as the ArcView project name.
To enter the characteristics of the OSBL network, use the lettered buttons on the far right of the

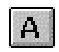


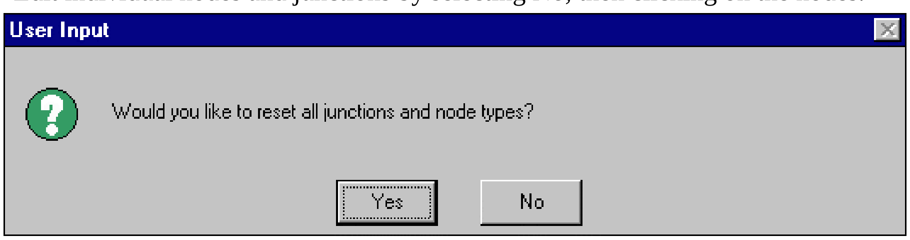
toolbar.

Certain steps in working with your OSBL unit can be done before completing the steps given for ISBL units. Other steps, however, require that *naUTilus* has already been run on all the ISBL units associated with your OSBL unit. In particular, the step which requires completion of all ISBL steps is that represented by the  button.

Entering OSBL data

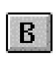
- Step A: Setting nodes and junctions

  To begin processing the OSBL unit, click on the  button in the toolbar. ArcView asks whether to reset all junctions and nodes. Select Yes if working with the unit for the first time. After this first time, select yes only if setting the nodes back to their defaults. Edit individual nodes and junctions by selecting No, then clicking on the nodes.

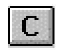
  

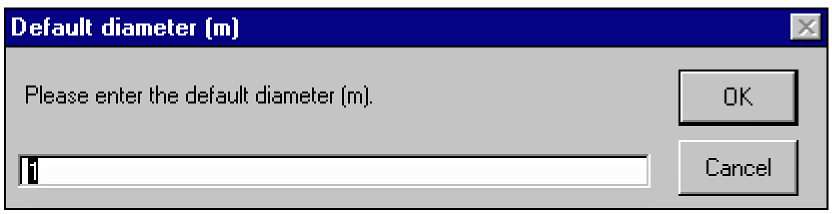
  The legend of the View shows the nodes color coded according to their type. A gray node indicates that there is a manhole present at that location. Purple signifies that no manhole exists. All start nodes (nodes at the initial end of flow lines) default to Node (no manhole). All internal nodes default to Node (manhole). When an internal node is selected it becomes a Junction (no manhole). Selecting a start node will toggle the node to indicate a manhole at that location. The outlet node is marked in a dark blue.

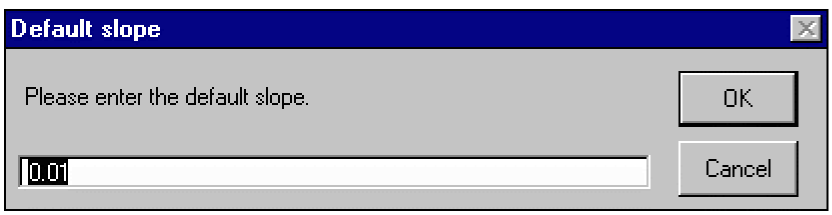
- Step B: Adjusting node numbering
  The numbering system in *naUTilus* does not account for the possibility of a manhole existing at the initial end of a flow line. When a manhole exists at the initial end of a flow line, ArcView accounts for it by adding a short branch to the system, to allow *naUTilus* to function on the network.

  Click on the  button. If the status of any of the purple nodes has been changed at any time, this step MUST be re-run. When this step has been completed, ArcView displays confirmation that the renumbering is complete

- Step C: Editing branch characteristics

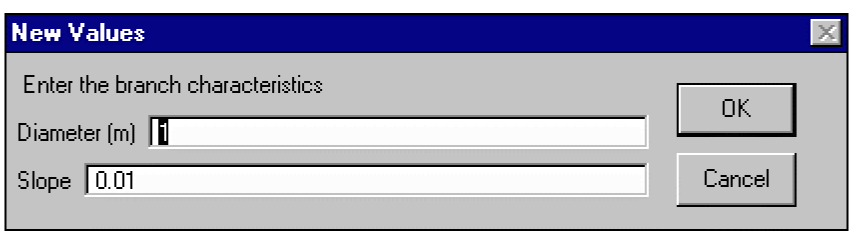
  To set branch characteristics, click on  the button. Select the default branch diameter and slope in the system. All branches will be set with these default characteristics.

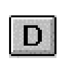




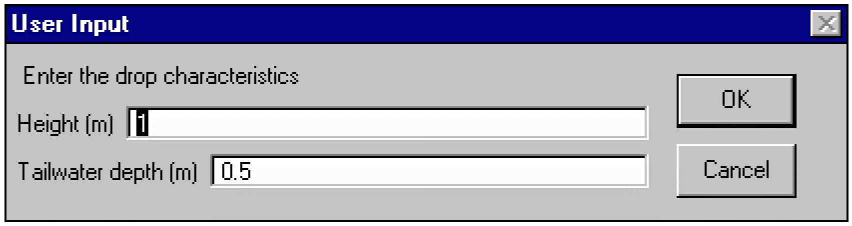
  Once the default characteristics are set, edit individual branch characteristics by clicking on individual branches.

  The values appearing in the input box are those currently assigned to the selected branch. Initially, these will be the defaults. Enter the new branch diameter in meters and the new branch slope. If either or both of the values are the desired values, click OK and move on to the next branch or to the next step.

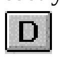


- Step D: Entering drop data
  To input drop data for the OSBL unit, click on the  button. Click on a branch where a drop occurs. As in the ISBL units, this branch has a drop at its terminal end.

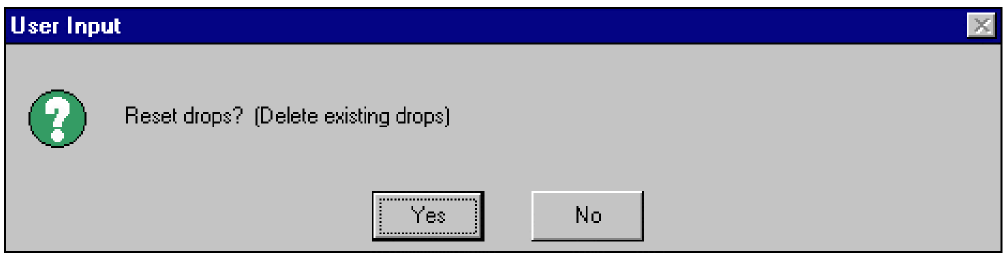
Enter the values of the drop height and the tailwater depth, both in meters. Enter these values and hit OK.



To reset your specified drops or, at a later time, choose to add new drops, click again on

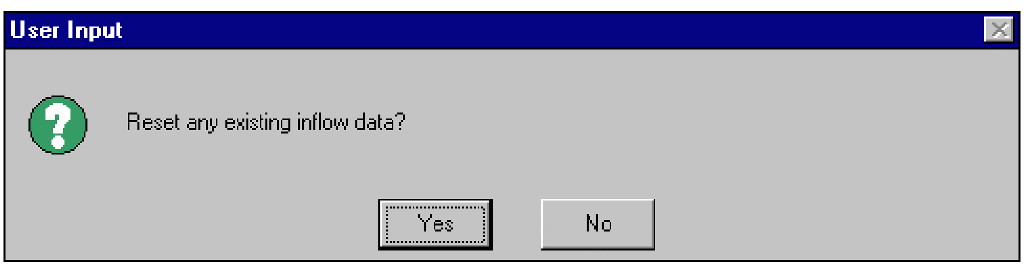


ArcView prompts whether to reset the existing drop data. To add new drops, select No. To reset the drop data, hit Yes.

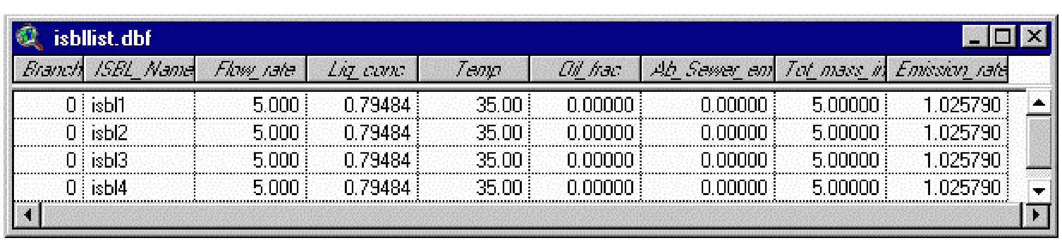


- Step E: Connecting your ISBL units/Specifying other flow
  To connect the ISBL units to the OSBL unit and to indicate if there is other flow to the OSBL network, click on the  button.

ArcView prompts whether to reset existing flow data. Choose the appropriate option.



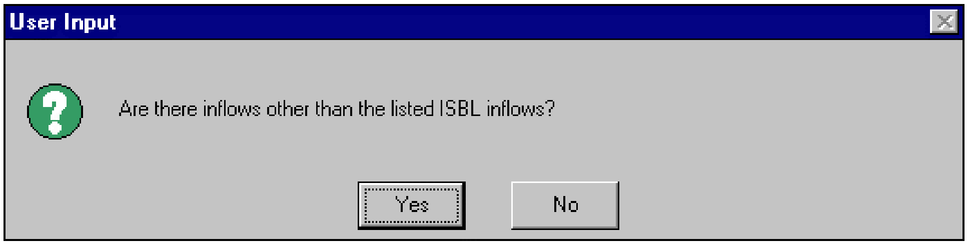
When the  button is first selected, ArcView will open the table "isbllist.dbf". This table shows if the ISBL units are already associated with any branch in the system, indicated under the column titled "branch". If they are not, a zero value appears in this column. Otherwise, the number which appears is the branch number to which the ISBL flows.



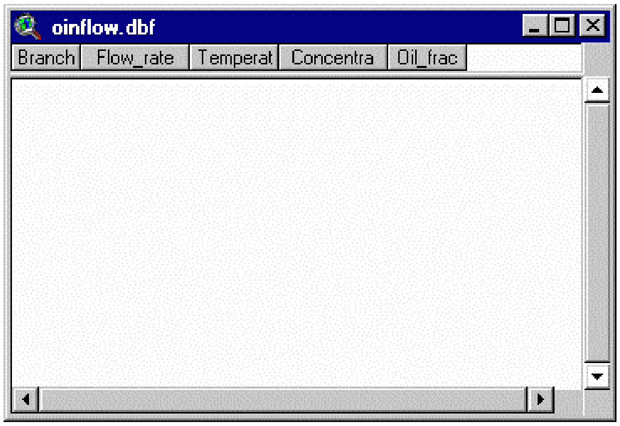
isbllist.dbf

Branch	ISBL_Name	Flow_rate	Liq_conc	Temp	Oil_frac	Ab_Sewer_em	Tot_mass_in	Emission_rate
0	isbl1	5.000	0.79484	35.00	0.00000	0.00000	5.00000	1.025790
0	isbl2	5.000	0.79484	35.00	0.00000	0.00000	5.00000	1.025790
0	isbl3	5.000	0.79484	35.00	0.00000	0.00000	5.00000	1.025790
0	isbl4	5.000	0.79484	35.00	0.00000	0.00000	5.00000	1.025790

ArcView also asks if other flows to the OSBL exist. If flows that do not originate in ISBL units, or if data on ISBL units that were not included in your ArcView project, exist, include them by selecting Yes.

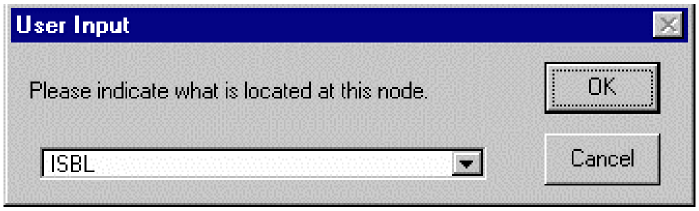


If Yes is selected, another table will open. This table will hold all data on non-ISBL inflows.

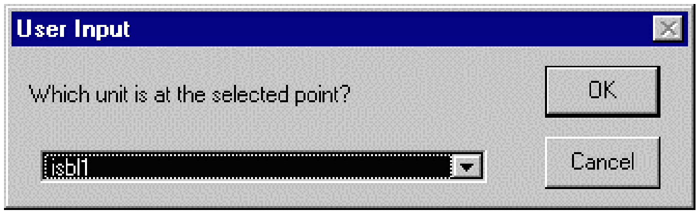


Next, click on a node where flow occurs. Note that as *naUTilus* assumes flow initiates at branches, not nodes, internal nodes are not presently allowed to be associated with ISBL or other inflow.

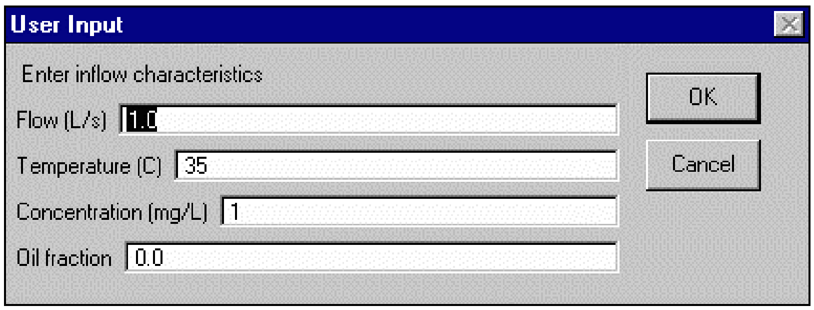
Once a valid point has been selected, indicate the type of inflow (ISBL or other). Select the appropriate option and hit OK.



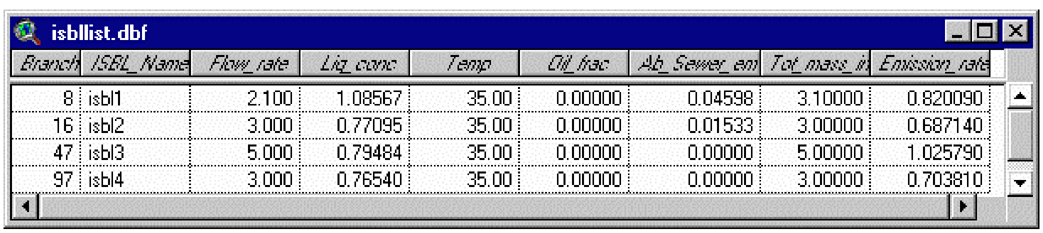
If ISBL flow is selected for the indicated point, ArcView presents a list of the ISBL units to select from. Select the ISBL unit which corresponds to the point and hit OK.



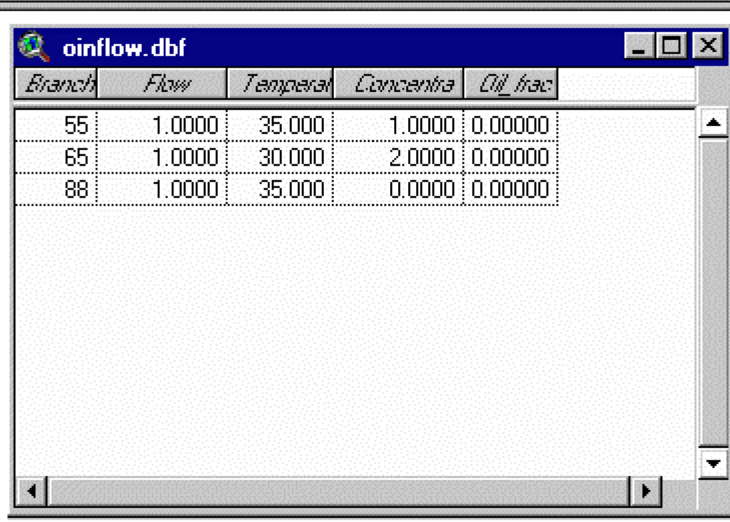
If other flow is selected, enter the flow data for that inflow. Enter the flow rate (L/s), temperature (^{0}C), concentration (mg/L), and the oil fraction. Hit OK.



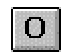
Complete connecting theISBL and OSBL units by repeating step E until all units and flows have been entered. Once complete, non-zero values should appear in the "branch" column of the isbllist.dbf table. Appropriate values should appear in the oinflow.dbf table.


isbllist.dbf

Branch	ISBL_Name	Flow_rate	Liq_conc	Temp	Oil_frac	Ab_Sewer_em	Tot_mass_i	Emission_rate
8	isbl1	2.100	1.08567	35.00	0.00000	0.04598	3.10000	0.820090
16	isbl2	3.000	0.77095	35.00	0.00000	0.01533	3.00000	0.687140
47	isbl3	5.000	0.79484	35.00	0.00000	0.00000	5.00000	1.025790
97	isbl4	3.000	0.76540	35.00	0.00000	0.00000	3.00000	0.703810

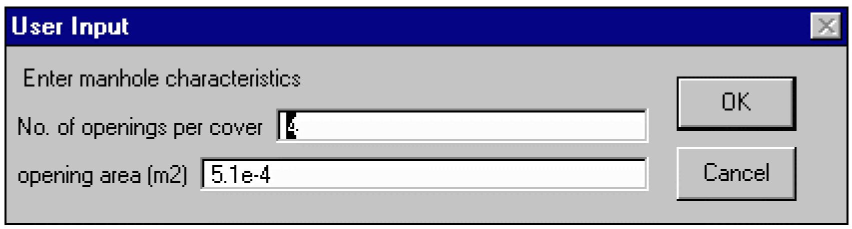

oinflow.dbf

Branch	Flow	Temperat	Concentra	Oil_frac
55	1.0000	35.000	1.0000	0.00000
65	1.0000	30.000	2.0000	0.00000
88	1.0000	35.000	0.0000	0.00000

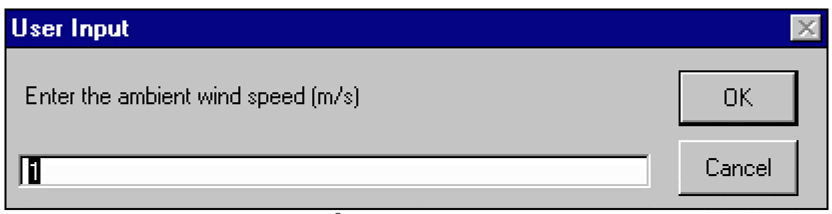
- **Running *naUTilus* (OSBL) and displaying emissions data**
  For the OSBL unit, creating a *naUTilus* input file, running *naUTilus*, and displaying the emissions at each node is done in one step. Click on the 

  button to start this step.

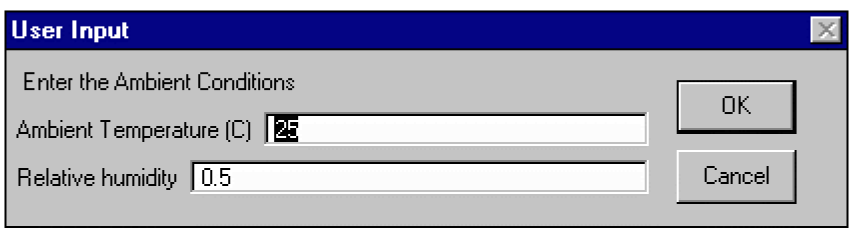
- Entering manhole characteristics:
  The first data required are the manhole characteristics. Enter the number of pick holes (openings) and the area of each opening (m^2).



- Entering ambient conditions:
  Enter your ambient wind speed (m/s) and hit OK.



  Enter the ambient temperature (^{o}C) and relative humidity as a decimal (i.e. 0.5 = 50%). Hit OK.

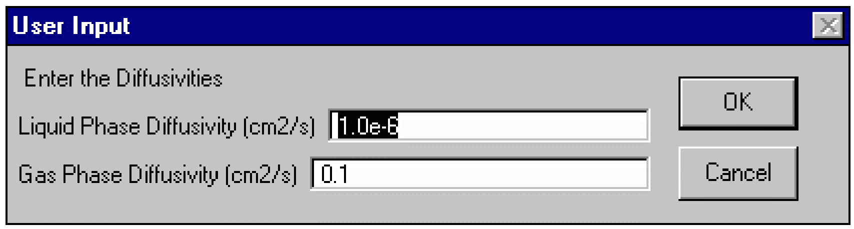


- Choosing a method to determine Henry's Law constants
  Choose the method of determining Henry's Constants throughout the OSBL system. For further documentation, see the *naUTilus* user documentation. The same four methods apply:

1. Method 1: H_c = exp(A-B/T)
   When using this method, *naUTilus* calculates the Henry's Law constant at each branch and node as a function of temperature using A and B constants. Required information: constants A and B.
2. Method 2: H_c, Antoine constants.
   In this method, Henry's constant is specified at a known temperature and related to Henry's Law constant other temperatures using a ratio of vapor pressure and Antoine constants. Required information: Antoine constants (A,B,C), Henry's Law constant at a known temperature.

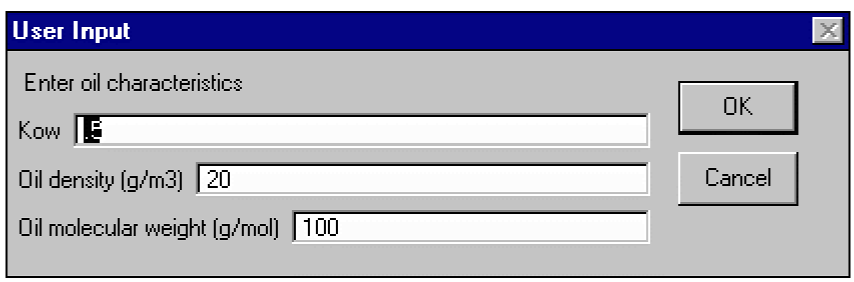
3. Method 3: H_c = Vp/sol using Antoine constants.
   Required information: Antoine constants (A,B,C), solubility, and molecular weight
4. Method 4: H_c = constant.
   Required information: Henry's Law constant

- Entering diffusivity coefficients:
  Enter the liquid phase and gas phase diffusivities (cm^2/s) and hit OK.



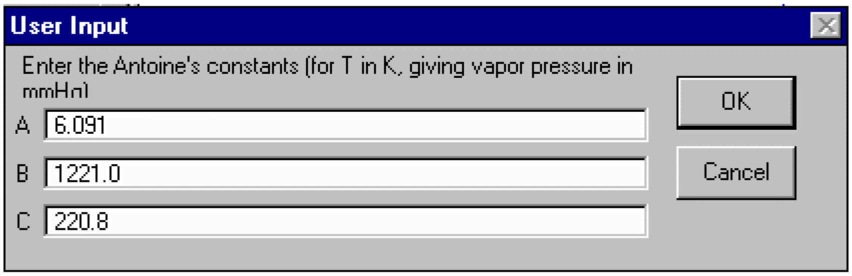
- Indicating the presence of oil:
  As in the ISBL case, three conditions are defined in *naUTilus* for the presence of oil in the system.
  - No oil present
  - Oil present only at the surface
  - Oil present and dispersed (not at the surface)

  If no oil is present, ArcView will move on to the next section. For oil at the surface or dispersed in the system, ArcView will prompt for an oil-water partitioning coefficient, oil density (g/m^3), and oil molecular weight (g/mol).

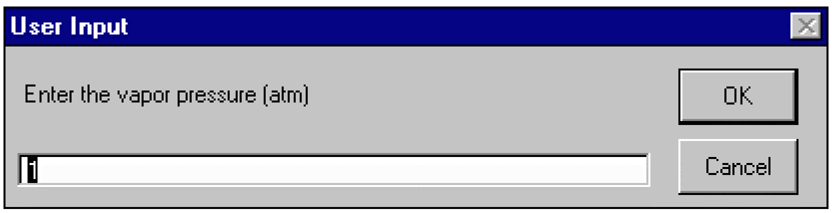


- Choosing the method to calculate vapor pressure (oil present)
  If no oil is present in the system, this step does not apply. If oil is present in the system, *naUTilus* has two options for calculating the vapor pressure: using Antoine constants or using a constant vapor pressure.

  Method 1 requires the Antoine constants (A,B,C) for temperatures in Kelvin. This gives vapor pressures in mmHg.



Method 2 requires a vapor pressure in atmospheres.



- Selecting a method to calculate reach emissions
  Again, this section is identical to that in the ISBL case and is discussed in the *naUTilus* user documentation (Olson, *et al.*, 1997). The two options for calculating reach emissions are:
  1. Parkhurst/Pomeroy
  2. Owens/Edwards/Gibbs

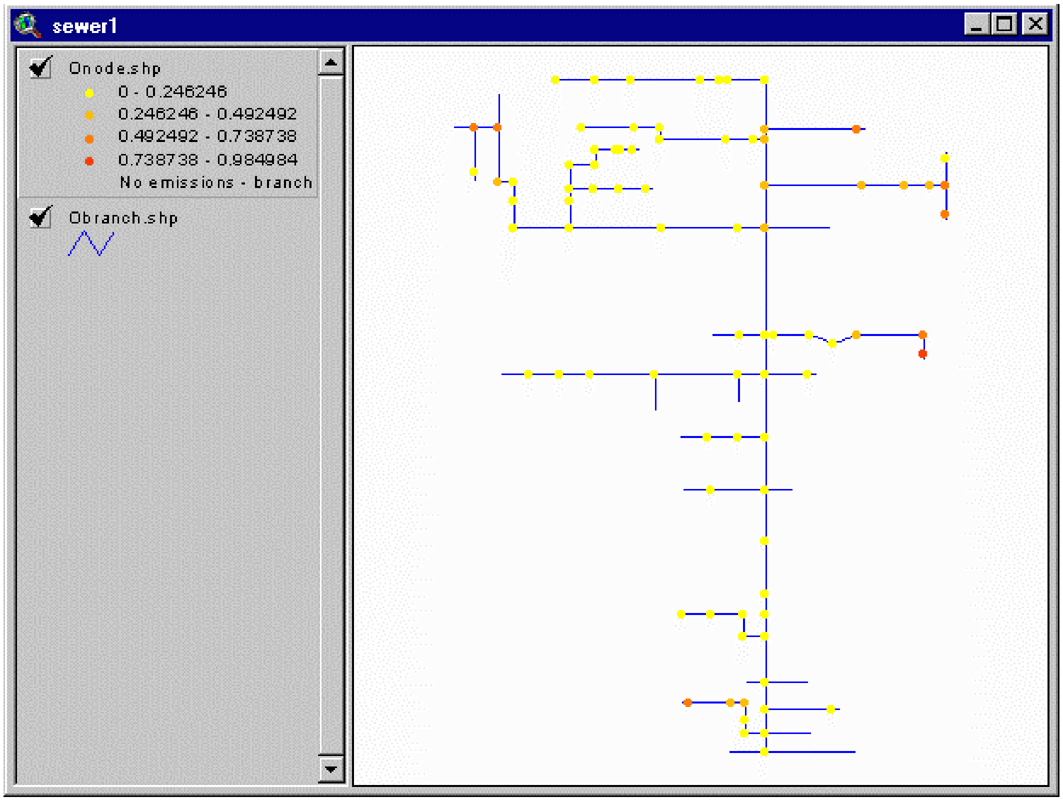
  Select the appropriate choice and hit OK.

- Choosing a method for calculating emissions at drop structures:
  Two options exist in *naUTilus* for the calculation of emissions at drop structures, as discussed in the *naUTilus* user documentation (Olson, *et al.*, 1997):
  1. Nakasone
  2. WATER8

  Select the appropriate choice and hit OK

  After the method is selected, hit OK. There will be a delay of approximately 10 seconds while ArcView calls and runs *naUTilus*. A DOS window will open and close, indicating that *naUTilus* has been run.

  Once *naUTilus* has been run, ArcView finds the resulting output and displays the results in your View window. An example of the output is shown.



- Tips on Displaying OSBL results
  The results shown upon executing naUTilus are default display options set by the ArcView project. Users familiar with ArcView can alter the display by editing the legend or using the Autolabel feature.

- Editing the legend:
  The legend editor allows the user to add or delete ranges from the default display of emissions. It also allows the user to display values for branches and values other than emissions for nodes.

  - Autolabeling features :
    The autolabel function is available from the theme menu. It allows the user to label features on the View with values from the feature attribute table.

Editing OSBL characteristics

Any of the sewer characteristics entered for OSBL may be edited using the buttons used for data input. As in the case of ISBL units, the user should be careful about resetting characteristics. As additional note for OSBL units, if any node types are changed in the OSBL unit, step B MUST be re-run.

Contacts

For questions regarding running *naUTilus through ArcView*, contact:
Cindy How
c.how@mail.utexas.edu

For questions regarding *naUTilus*, contact:
Dr. Richard Corsi
The University of Texas at Austin
corsi@mail.utexas.edu
(512)475-8617

References

Olson, D.A., S. Varma, and R.L. Corsi. 1997(b). naUTilus: A Model for Predicting Chemical HAP Emissions from Industrial Sewers; I. User Documentation, II. Technical Documentation, report to the United States Environmental Protection Agency.

Exercise: Running *naUTilus* from ArcView
Center for Research in Water Resources &
Center for Energy Studies - UT Program in Air Resources Engineering
The University of Texas at Austin

Using ArcView as a User Interface for *naUTilus*

By Cindy How

c.how@mail.utexas.edu

Table of Contents

Goals of the exercise

- To provide a step by step guideline for processing a simple network (1 ISBL, 1 OSBL) in ArcView
- To complement the User Documentation
- To provide simple hypothetical ISBL and OSBL units to be used as a test case
- To run *naUTilus* from ArcView for the test case

Computer and Data Requirements

Software:

This project was set up to run on a PC system running ArcView (version 3.0 or higher). The procedure also requires Arc/Info. You will also need the *naUTilus* executable files.

<table>
<tr><th colspan="3">naUTilus Executable Files:</th></tr>
<tr><td colspan="3">1. isbl.exe
2. osbl.exe</td></tr>
<tr><th colspan="3">Data Files:</th></tr>
<tr><th>ISBL</th><th>OSBL</th><th>Other (required by ArcView project)</th></tr>
<tr><td>1. ibranch.shp
2. ibranch.shx
3. ibranch.dbf
4. inodes.shp
5. inodes.shx
6. inodes.dbf</td><td>1. obranch.shp
2. obranch.shx
3. obranch.dbf
4. onode.shp
5. onode.shx
6. onode.dbf</td><td>1. nautilus.apr
(ArcView project file)
2. osblnd.avl
3. sewerleg.avl
4. sewerlg2.avl</td></tr>
</table>

Obtaining the Files:

The *naUTilus* executable files are not presently available for general release. Please contact me at c.how@mail.utexas.edu about obtaining files till all files are publicly available.

Introduction

This exercise was created to complement to User Documentation which describes the general procedure for running *naUTilus* from ArcView. This exercise provides step by step instruction for an example sewer network (1 ISBL unit, 1 OSBL unit). It assumes that the ISBL and OSBL units have already been digitized, saved as DXF files, and converted to theme files in Arc/Info.

Procedure

1. Getting the necessary files

Create a directory called OSBL on your computer. Place the following files in the OSBL directory:

1. nautilus.apr
2. osbl.exe
3. isbl.exe
4. osblnd.avl
5. sewerleg.avl
6. sewerlg2.avl
7. obranch.shp
8. obranch.shx
9. obranch.dbf
10. onode.shp
11. onode.shx
12. onode.dbf

Within the OSBL directory, create a subdirectory called ISBL. In this directory, place the following files:

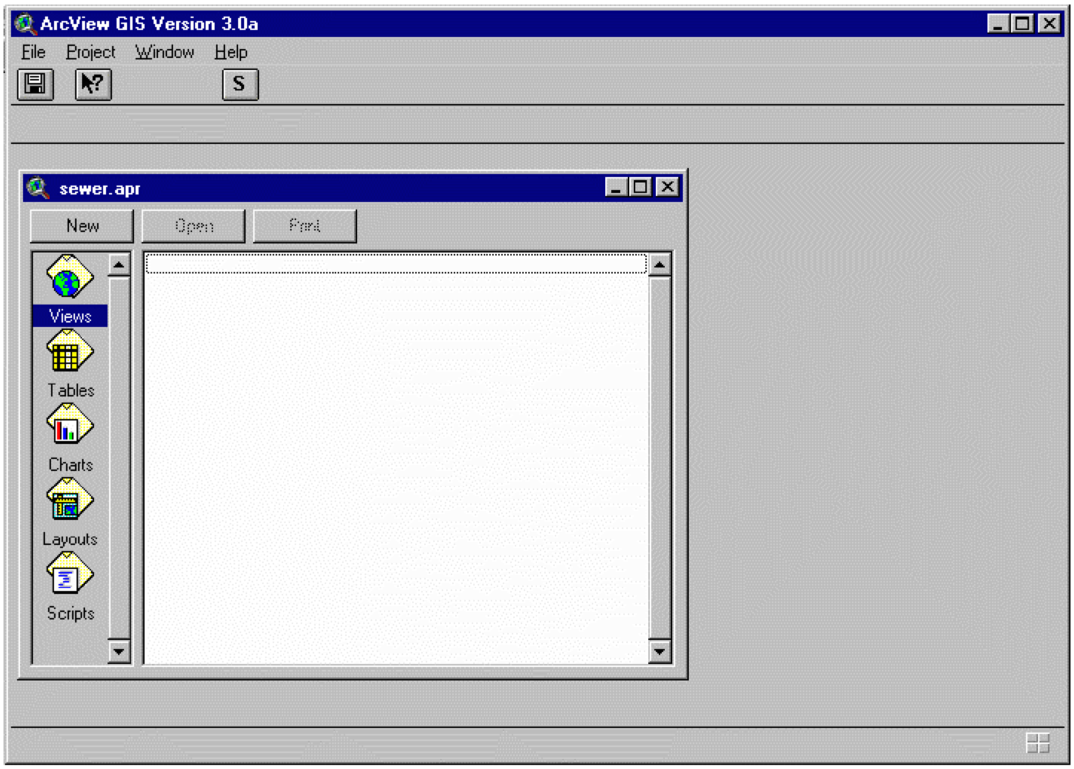
1. ibranch.shp
2. ibranch.shx
3. ibranch.dbf
4. inodes.shp
5. inodes.shx
6. inodes.dbf

Note that the names of the directories do not need to be OSBL and ISBL. These names are used for simplicity and uniformity in the exercise. However, the names of the theme files (inodes.*, ibranch.*, onode.*, obranch.*) do need to be as specified.

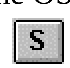
2. Open the ArcView project

Once your files are set up, start the application ArcView. Open the project, nautilus.apr. From the menu bar, go to File/Save as. **Save** your project as sewer.apr. (This name is used for simplicity and uniformity in the exercise. For original networks, name the project as appropriate to the OSBL unit. The name given to the project will also be given to the OSBL view window.)

When your project is opened, it will appear as below:



3. Initiating the network files

Now that the project file is opened and saved, you will want to bring the OSBL and ISBL theme files into your project. This is done by hitting the 

button found near the top left of the screen. It should appear as seen here, beneath the Help item of the menu bar.

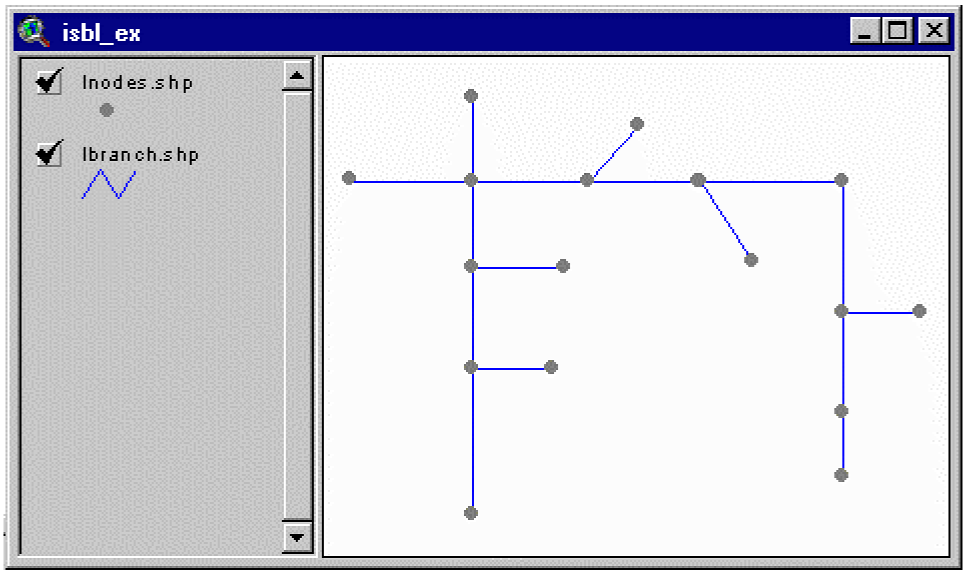
This step not only finds the network theme files in your directories, it also creates files needed throughout the project. A "map" to these files is also created in this step. Note that this step MUST be run for each new project. Other files may be manually added to the project, but this step is essential for all scripts to run properly.

From your project window, click on the View icon.

You can see the views "sewer" and "isbl" have been added to your project. Click on the Tables icon and you can see the table isbllist.dbf. Several other files are created in this step, but are not opened in your project till they are needed.

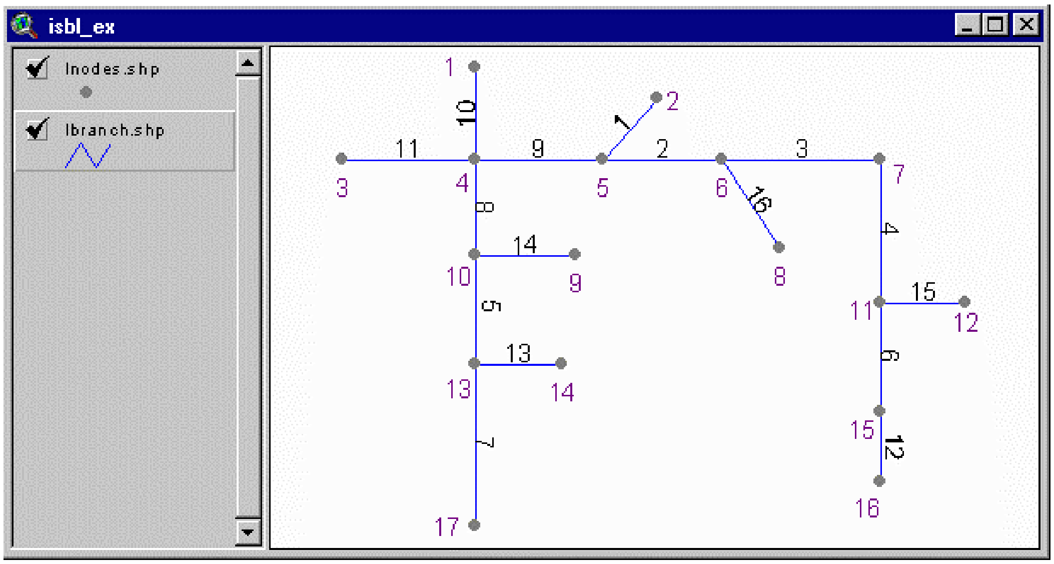
4. Working with the ISBL network

Click on the View icon and select the view called "isbl". Click on Open. This opens the view window describing your ISBL network. Initially, your ISBL network should look like this:

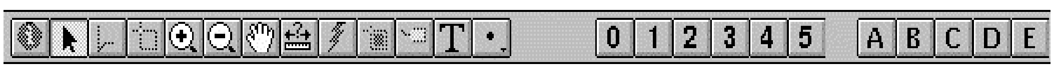


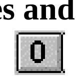
Once your view window is open, there are 6 steps you will need to perform to input data on your ISBL unit. These six steps are run by hitting the buttons numbered from 0 - 5 in the toolbar.

To help you work with this network, the nodes and branches have been numbered as follows:



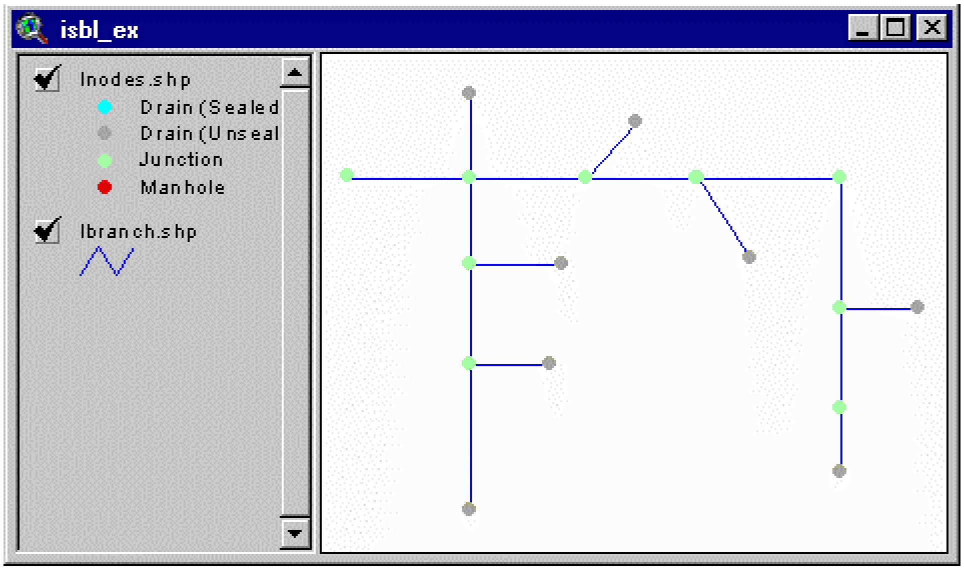
Branch numbers appear in black and node numbers appear in purple



a. **Setting nodes and junctions**
Click on the  button.

This is the first time the network is being edited so you will be asked to choose a default for you nodes. Of the three options (all nodes are drains, all nodes are junctions, or internal nodes are junctions and initial nodes are drains), select option 3 and hit okay.

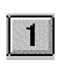
Notice that the legend for the theme "inodes" has changed. Four different kinds of nodes are listed in the legend (at the left portion of the view window). Look at your ISBL network. Notice that all initial nodes are gray, indicating unsealed drains. All of the internal nodes should be green, indicating they are junctions.



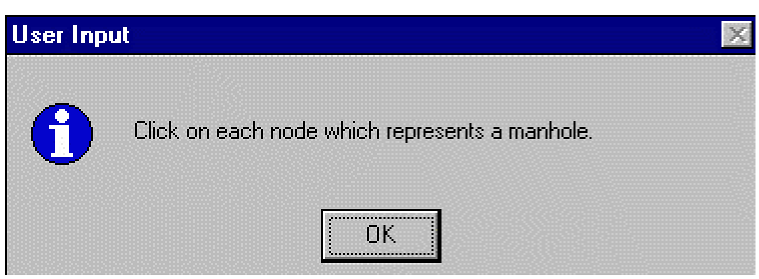
Now you are prompted to click on the nodes you wish to edit. Click on a node and notice that it changes between green (junction) and drain (gray). Click on the same node again and see that it toggles back to its original default setting. Click on node 15. Do not worry about the sealed/unsealed status of the drains. You will be able to change drains to sealed drains at a later point.

If you should wish to change a drain's status (drain/junction) any time after moving on in the exercise you will be asked if you want to reset the drain data. If you select yes, you will start from the beginning of this step again. If you select no, you will be prompted to select nodes you wish to edit, as in the step discussed immediately above.

b. **Setting manholes**

Click on the  button.

You are prompted to click on any nodes which represent manholes.

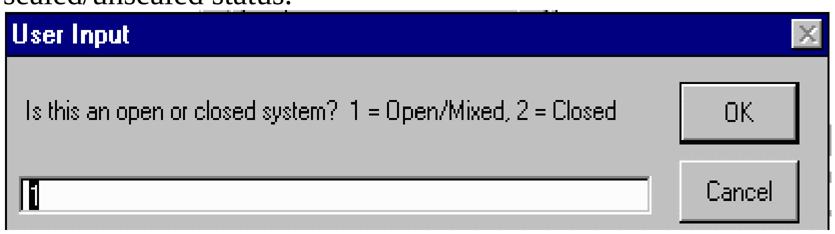


Click on a node. Notice that it turns red. Look at the legend for the node theme. It indicates that a red node marks a manhole. Click on that same node again. You are prompted to indicate if this node is a junction or a drain. Select an appropriate option. Click on node 4.

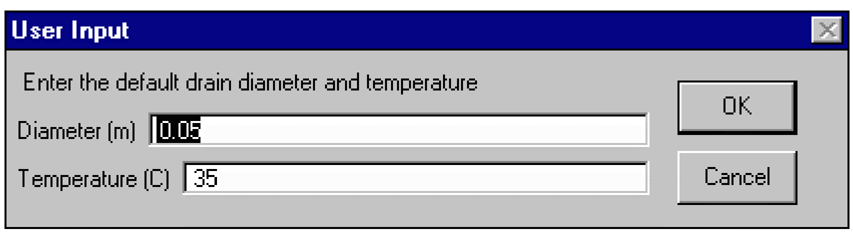
Editing drains

Click on  the button.

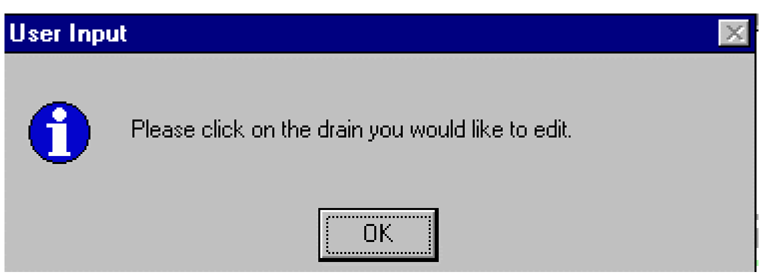
You are asked to indicate if your system is Open/Mixed or Closed. Select Open/Mixed. This sets all of your drains as unsealed. For either selection, you will be able to select individual drains and change their sealed/unsealed status.



Enter a default drain diameter of 0.05 meters and a default temperature of 35 ^{o}C. Click OK.

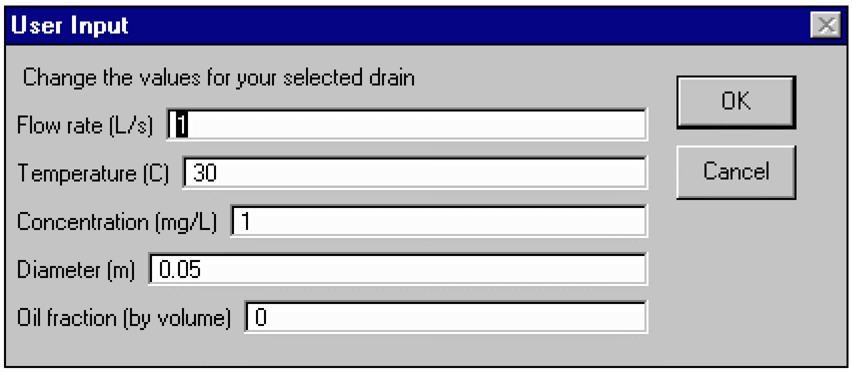


You are prompted to click on a drain you would like to edit.



Click on node 14. You are given two options: change the sealed/unsealed status or enter flow data. Select "Change sealed/unsealed". You will be asked to indicate what kind of drain you want to change it to. Select sealed and hit OK. Notice that the color of node 14 has changed from gray to blue, indicating that it is now a sealed drain.

You will now enter flow data. Click on node 17. Select the option "Enter flow data". Enter a flow rate of 1 L/s, a temperature of 30 ^{o}C, a concentration of 1 mg/L, a diameter of 0.05 meters, and an oil fraction of 0.



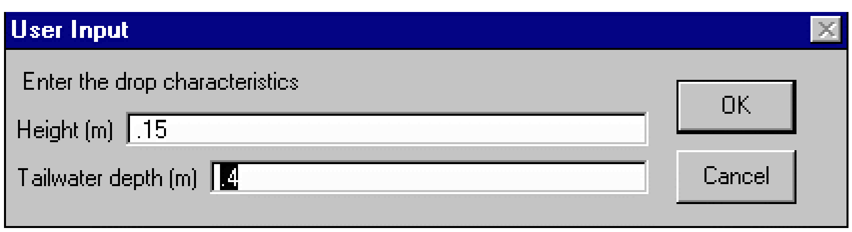
Click on node 15. Select "Enter flow data" and enter a flow rate of 1.5 L/s, a temperature of 30 ^{o}C, a concentration of 0 mg/L, a diameter of 0.05 meters, and an oil fraction of 0.

c. **Entering drop data**

Click on the 

button.

Click on branch 8. Enter a drop height of 0.15 meters and a tailwater depth of 0.4 meters, then hit OK.



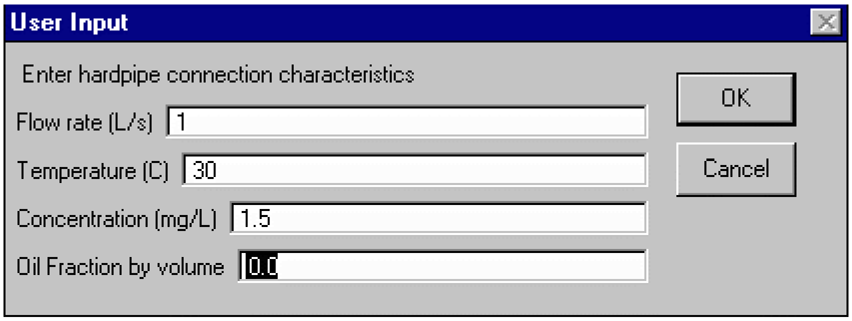
Repeat the above for branches 9 and 10.

d. **Enter hard pipe connections**

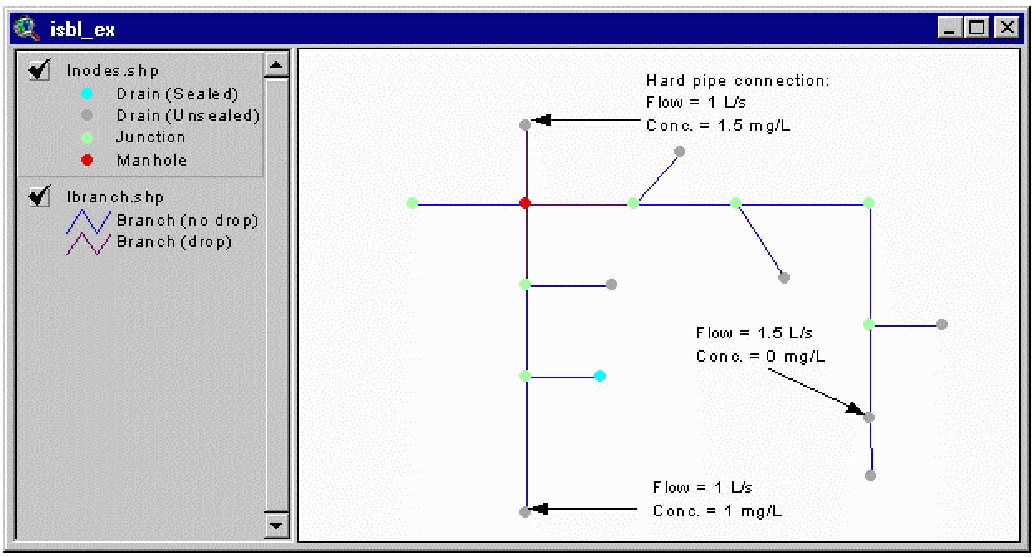
Click on the 

button.

Click OK at the prompt. Click on node 1 and enter the following data: flow = 1 L/s, temperature = 30 ^{o}C, concentration = 1.5 mg/L, oil fraction = 0.0.



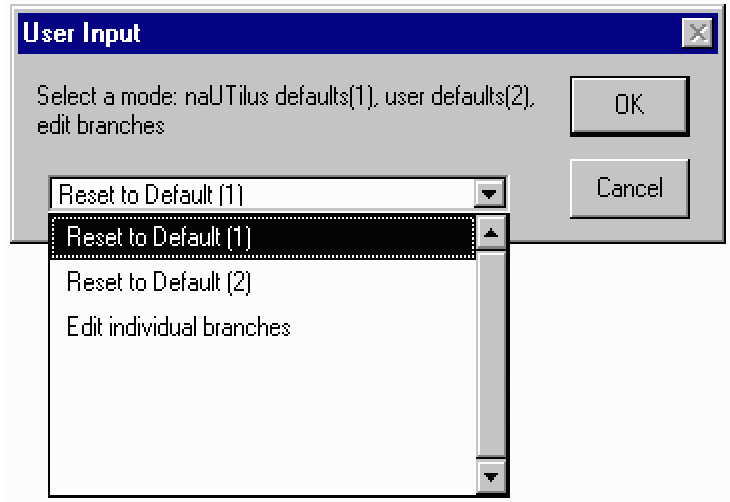
You have now completed entering the data on the flows to you ISBL system, which can be summarized by the following graphic:



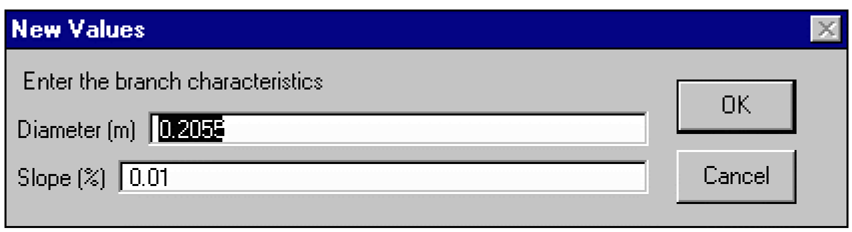
e. **Editing branches**

Click on  the button.

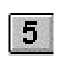
Select the option "Reset to Default (1)". When prompted for a default slope value, enter 0.01 and hit OK. Hit OK at the next prompt.



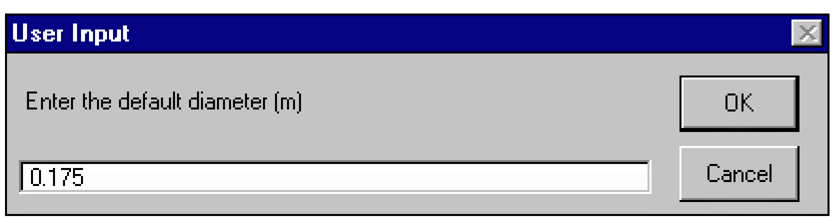
Click on branch 11. You can see that the diameter of that branch is 0.2055 meters. Click on any of the other branches. You can see that the diameter for other branches is 0.1534 meters. These are the values *naUTilus* uses



(0.2055 for branches directly downstream from a manhole and 0.1534 for other branches).

Click on the 

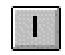
button again. This time, select the option "Reset to Default (2)". When prompted, enter a default diameter of 0.175 meters and hit OK. Enter a default slope value of 0.005 % and hit OK. Hit OK at the next prompt.



Click on any branch. You can see that all branches have a diameter of 0.175 meters and a slope of 0.005%. Click on branch 11 and change the diameter to 0.2 meters. Hit OK. If you click on branch 11 again, you will see this new value listed as the diameter.

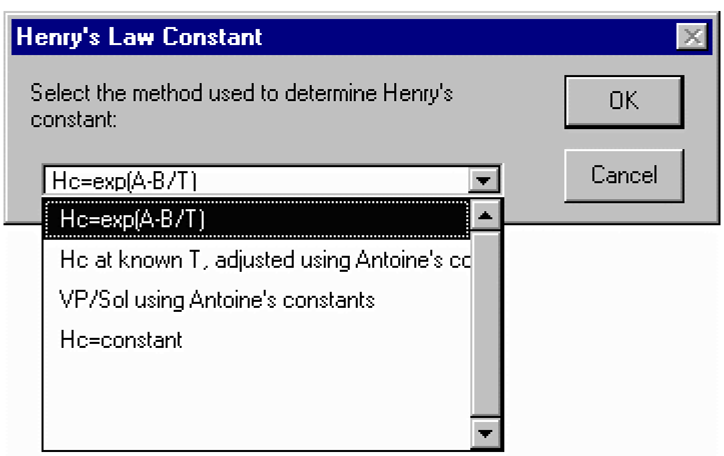
You are now ready to run *naUTilus* on your ISBL unit.

5. Running *naUTilus* for ISBL

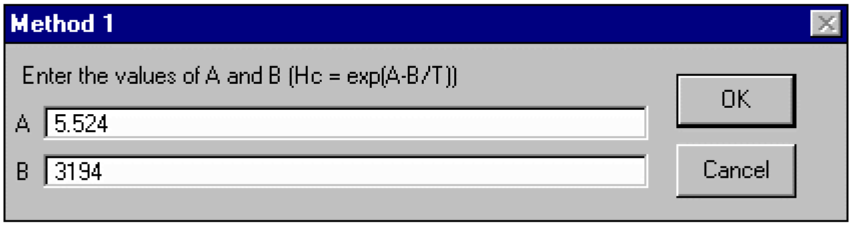
With your ISBL view window active, click on the  button.
We will use benzene as the VOC analyzed in this problem. The chemical data for benzene you will need for this example are provided in this exercise. The data needed are the following:

- A and B for van't Hoff's equation (calculation of the Henry's Law Constant)
  A = 5.524, B = 3194
- D_l, D_g (liquid and gas phase diffusivities)
  $D_l = 0.00001$, $D_g = 0.09$

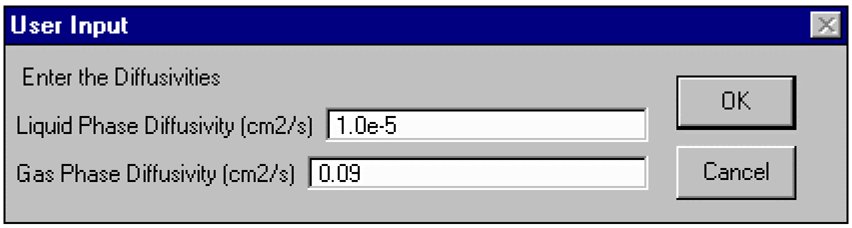
The first thing ArcView will prompt you for is the method of calculating the Henry's law constant. Select the method one (Hc=exp(A-B/T)) and hit OK.



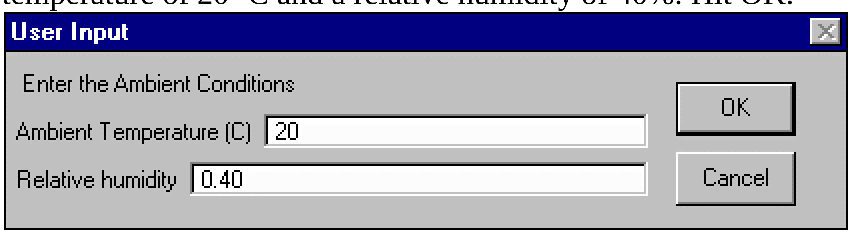
Enter the values of A and B listed above. Hit OK.



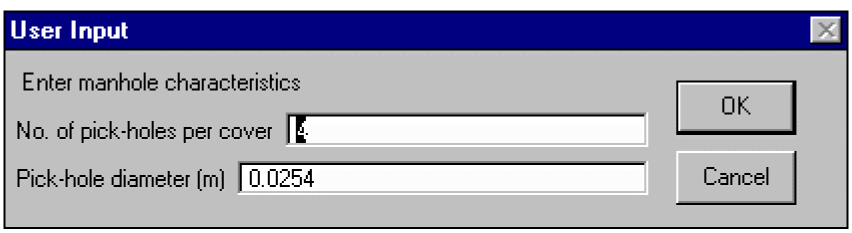
Next you will enter the values for D_l and D_g.



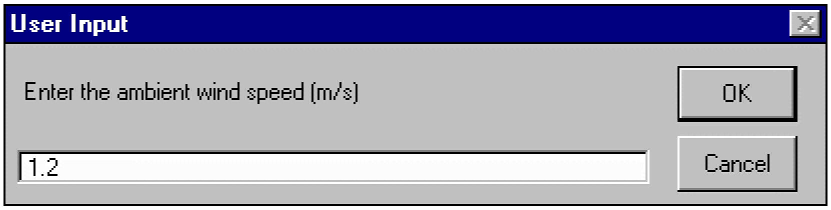
You will then be prompted to enter some ambient conditions. Enter an ambient temperature of 20 ^{o}C and a relative humidity of 40%. Hit OK.



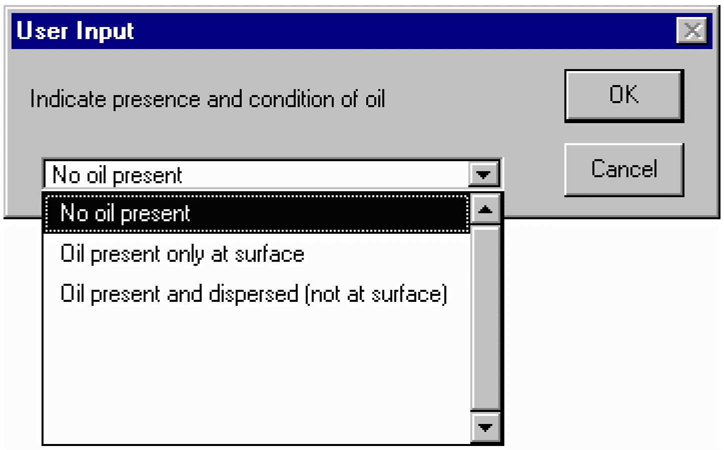
When you are prompted for the number of pickholes per manhole cover and the pickhole diameter, accept the default values given. Hit OK.



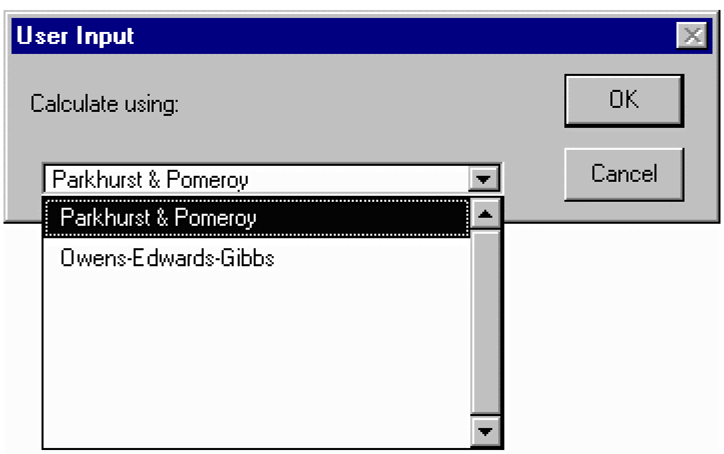
When prompted for the ambient wind speed, enter a value of 1.2 meters per second. Hit OK.



You will then be asked to indicate if oil is present in your system. We will assume that there is no oil in our case. Select the first option (No oil present) and hit OK.

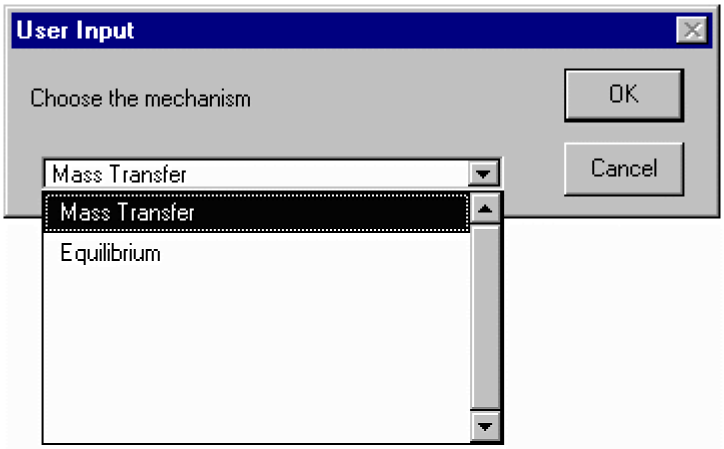


ArcView will then ask you to choose the relationship you would like to use for

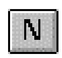


calculating . Select method one (Parkhurst/Pomeroy) and hit OK.

Next you will indicate if you believe the mass transfer in the system is kinetically limited or equilibrium limited. We will assume our system is kinetically limited.



Select the first option (Mass Transfer) and hit OK.

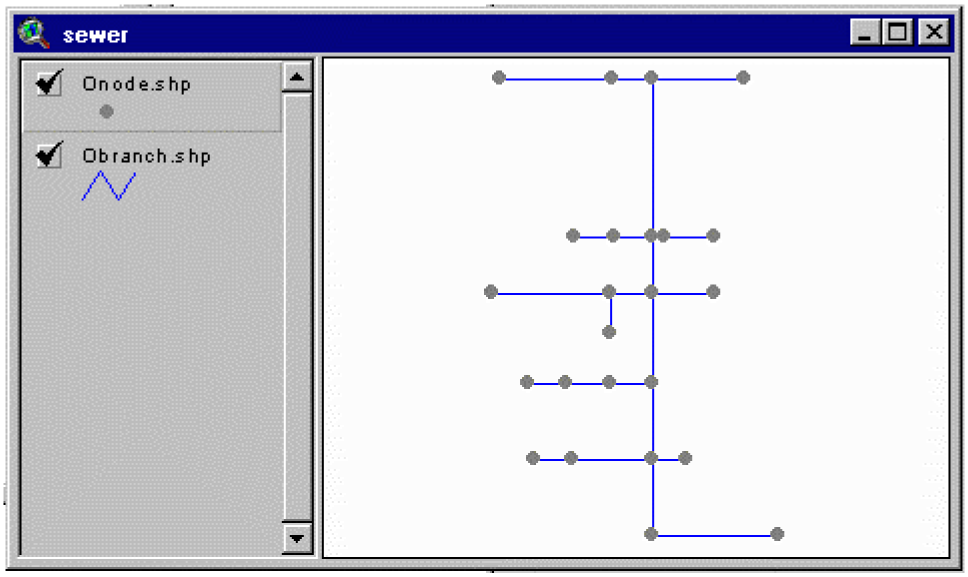
You are now done creating the ISBL *naUTilus* input file. The next step is to run *naUTilus*. This is done by selecting the 

button.

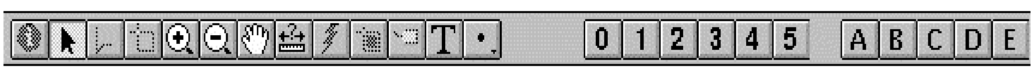
This is a delayed run, so there will be approximately 10 seconds before *naUTilus* is run. When it is run, you will see a DOS window open and run several commands. This DOS window should close when the commands have been run. ISBL *naUTilus* has been run. Look at the output files. You can look at it by opening it in Notepad. There are two ISBL output files, ISBL.OUT and ISBLOUT.TXT. ISBL.OUT lists data for each branch in your system. The other output file, ISBLOUT.TXT, is a small text file with the data needed for OSBL input. Copies of both output files are included at the end of this document.
You have now completed working with the ISBL in your system.

6. Working with the OSBL network

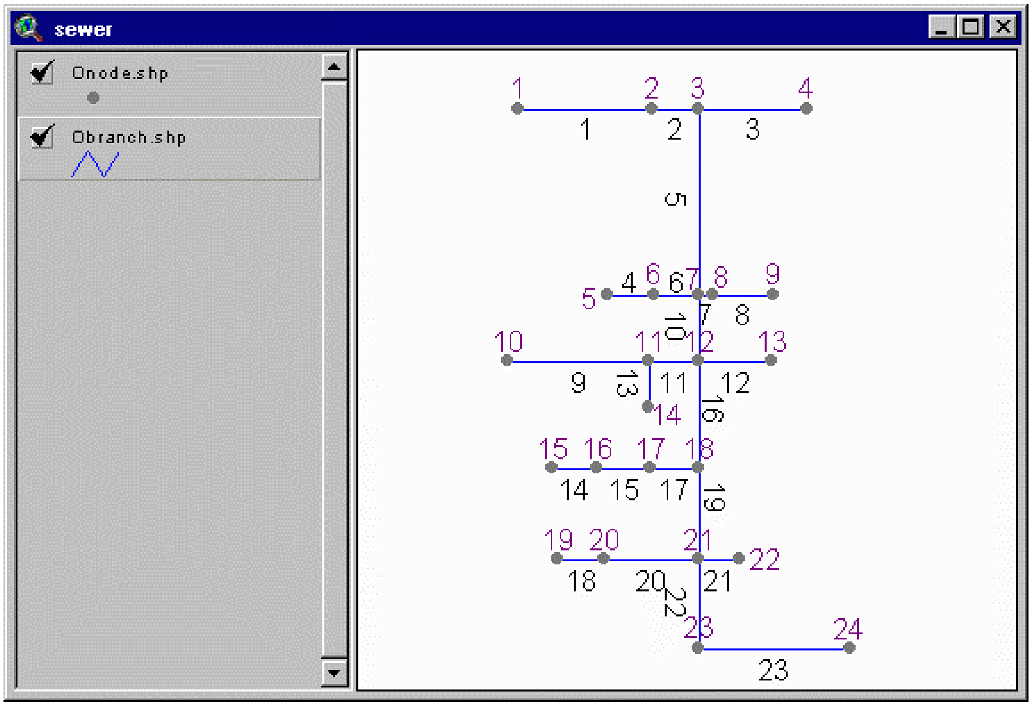
The steps to work with the OSBL network are very similar to those for the ISBL network. Open your OSBL view window. It should look like this:



The buttons to work with your OSBL network are to the right of those you used for the ISBL network and are labeled from A through E.

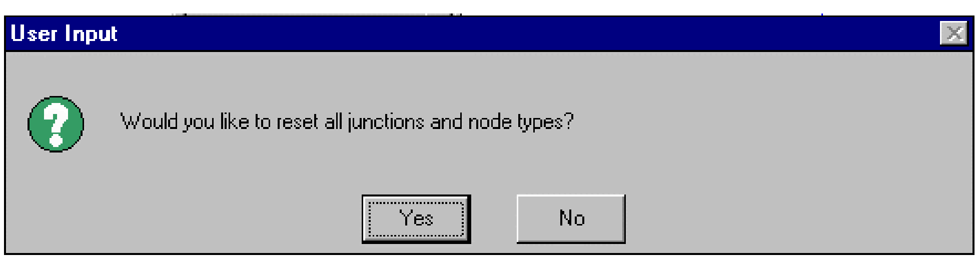


To help you work with the OSBL network, a numbered diagram of the nodes and branches has been provided. Node numbers appear in purple and branch numbers appear in black.

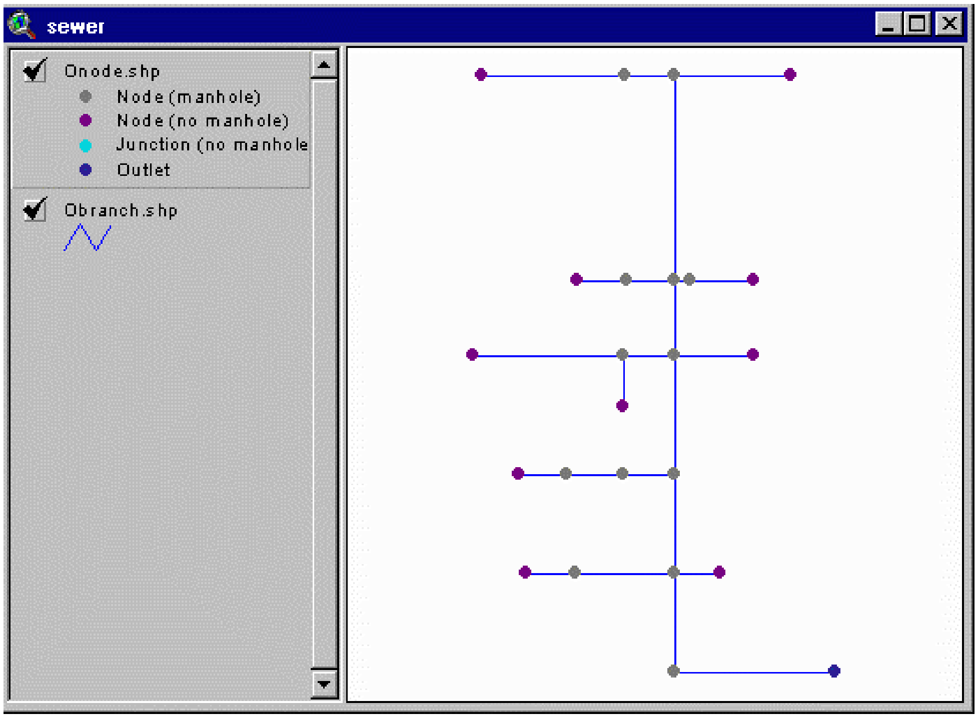


a. **Setting nodes, junctions, and manholes**
Click on the A button.

This is the first time you have worked with the network, but you will be asked if you want to reset all junction and node types. Click on Yes.



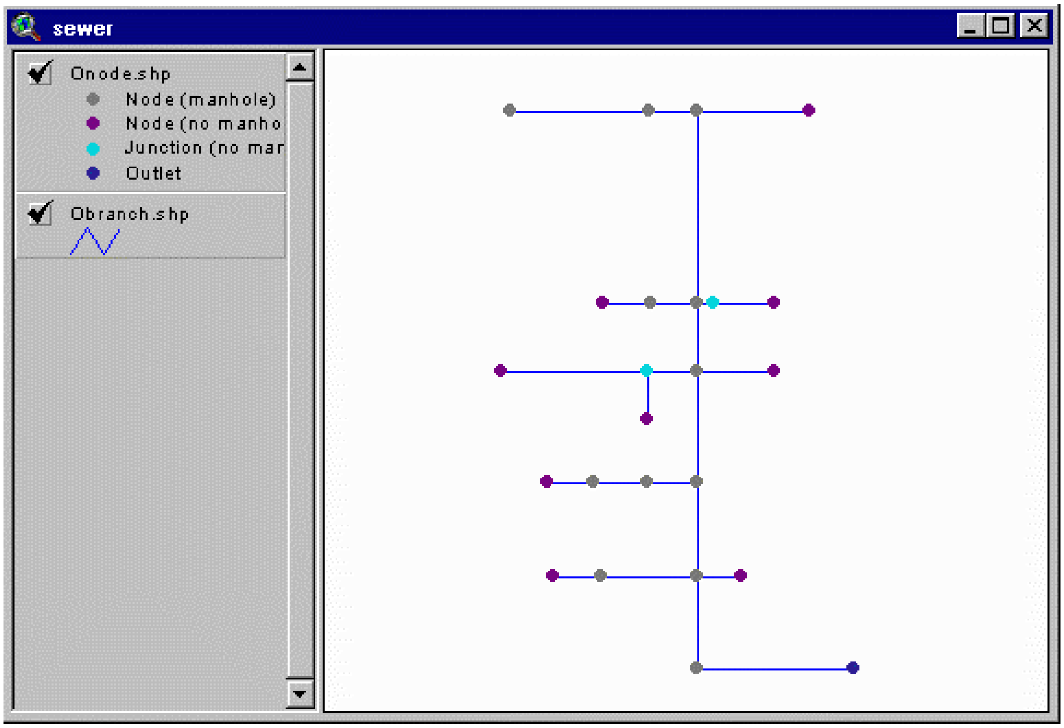
Your OSBL network should now look like this:



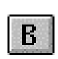
Notice that the outlet (located at the bottom right of the image) is colored in dark blue, and nodes at initial branch ends are purple (no manholes). All other nodes are gray, indicating a manhole location.
You are prompted to click on any node to edit the node type. Click on node number 10. You will see it change from purple to gray, indicating a manhole is located at node 10. Click on node 10 again. You will see it toggle back to purple, indicating there is no manhole at that location.

Click on nodes 1, 8, and 11. The resulting OSBL should look like this:

There is now a manhole at node one and junctions (with no manholes) at nodes 8 and 11.

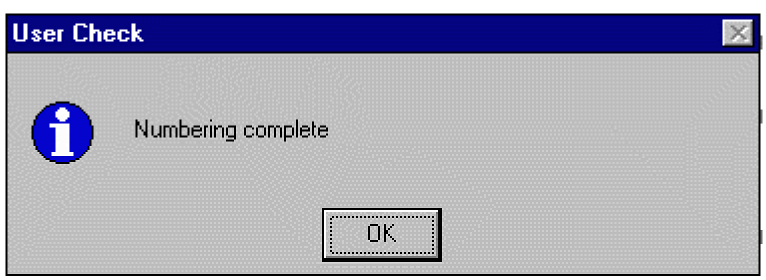


b. **Assigning *naUTilus* numbering**

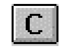
To assign all nodes numbers consistent with the *naUTilus* numbering system, click on the 

button.

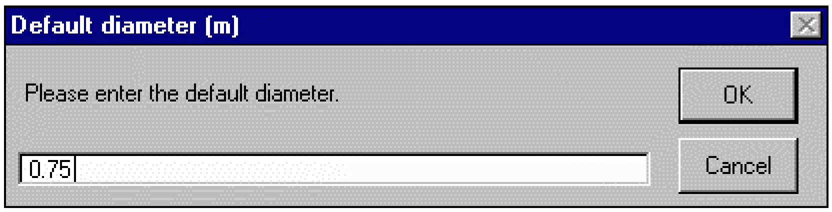
When the numbering is complete, you will see the following message. Click OK.



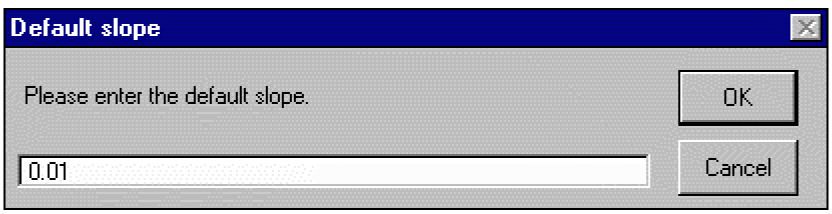
c. **Editing branches**

Next you will enter the default branch characteristics. Click on the  button.

When prompted for a default branch diameter, enter the value 0.75 meters.

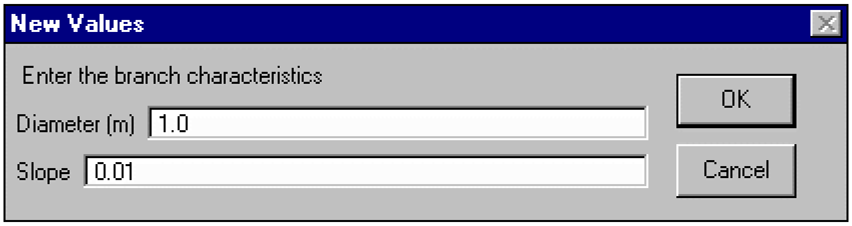


Select a default branch slope of 0.01%.



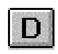
You will be prompted to click on the branches you wish to edit. Click OK.

Click on branch 5. Replace the default value for diameter (0.75 meter) with the value 1.0 meter. Leave the default branch slope. Click OK.



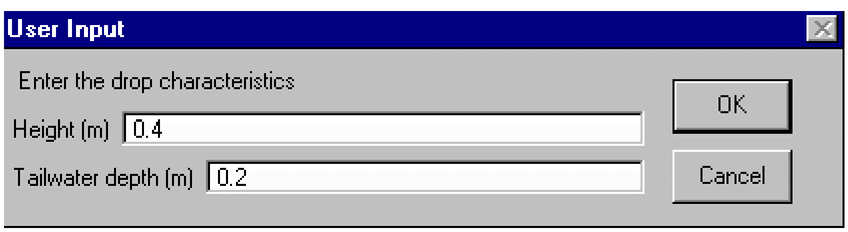
Repeat this step for branches 10, 16, 19, 22, and 23. These are the main branches of the system (laterals).

d. **Entering drop data**

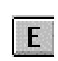
To enter drop data, click on the 

button. You will see a message to click on a branch where a drop occurs. Click OK.

We will assume there is a drop structure at node 21. Click on branch 19. When prompted enter a drop height of 0.4 meter and a tailwater depth of 0.2 meter. Then click OK.

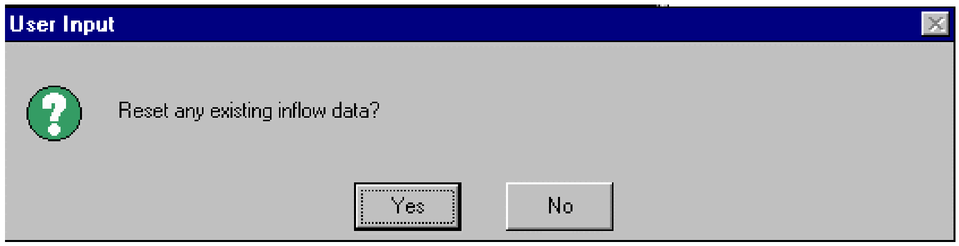


Repeat the above procedure for branches 20 and 21.

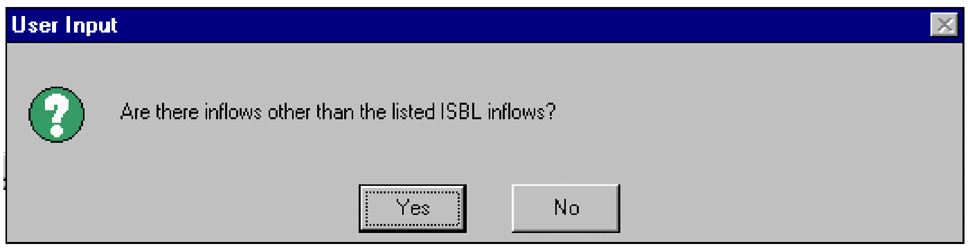
e. **Entering flow data**
The last step in describing your OSBL network is to connect the ISBL network and enter other flow data. To do this, click on the 

button.

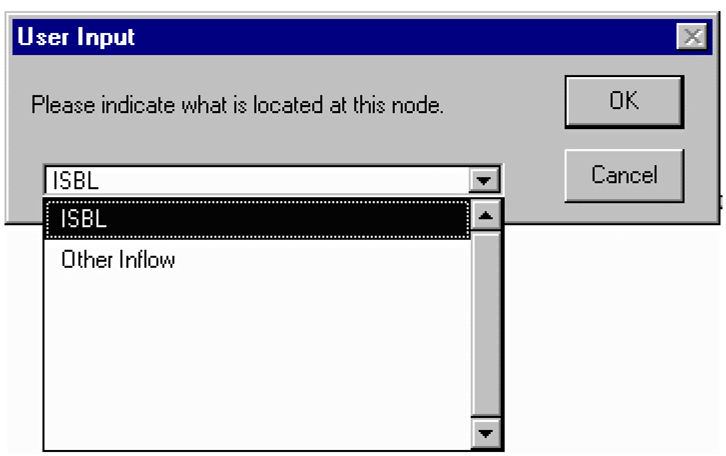
This will open the table ISBLLIST.DBF. You will also see a message asking if you would like to reset any flow data. Select Yes.



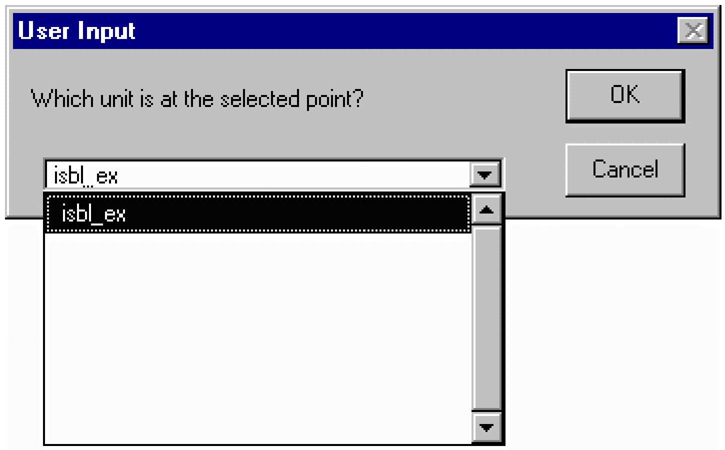
You will then be asked if there are inflows other than the ISBL. Select Yes. This will open the table OINFLOW.DBF. It is initially an empty table.



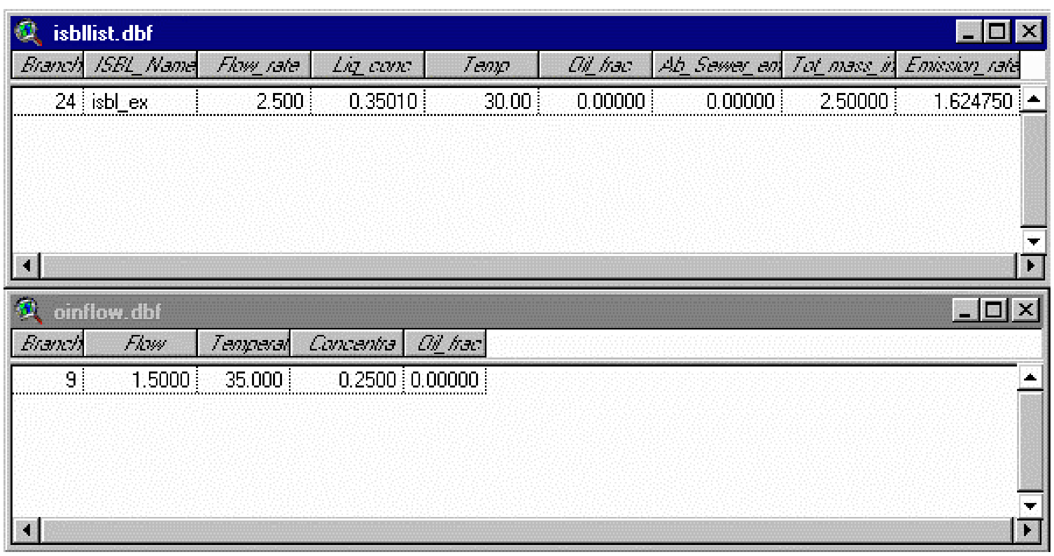
Click on node 1. ArcView will ask you to indicate the type of flow located at node 1. Select ISBL



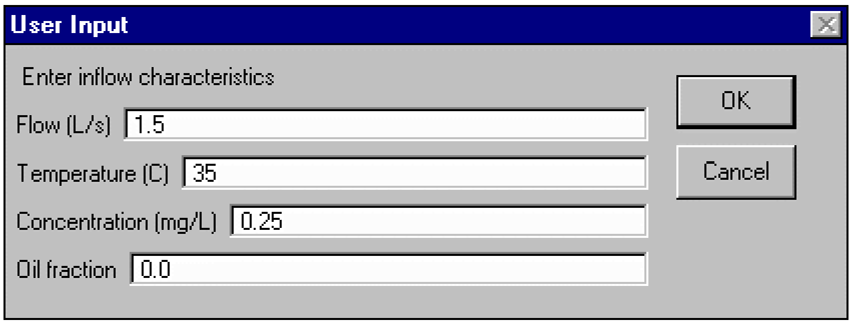
You are now asked to indicate which ISBL is located at node 1. We only have one ISBL unit (ISBL_EX) associated with this OSBL. Click OK.



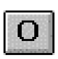
Now, if you look at the table ISBLLIST.GIF, the data output from running *naUTilus* on the ISBL_EX unit is listed in the table. The ISBL unit has been connected to the OSBL.



Click on node 10. Again, you will be prompted to indicate the type of flow at the node. This time, select Other Inflow. Enter the following flow characteristics: flow = 1.5 L/s, temperature = 35 ^{o}C, concentration = 0.25 mg/L, and oil fraction = 0.0. Then hit OK. This data will now appear in the table OINFLOW.DBF.

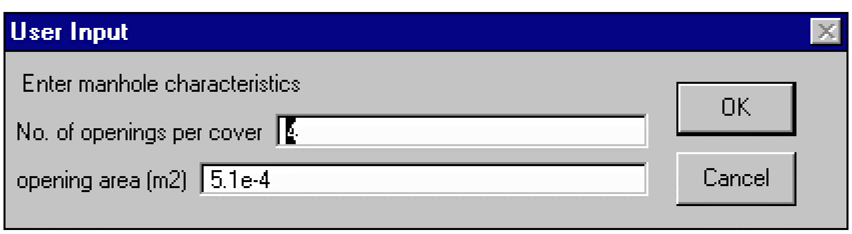


7. Running *naUTilus* for OSBL

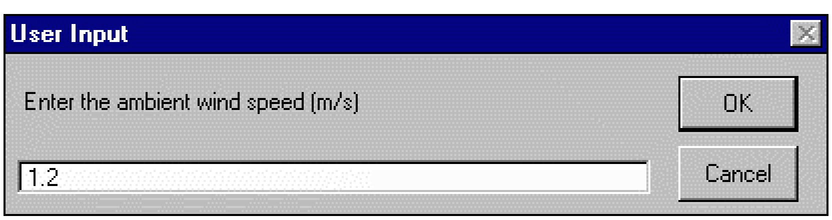
You are now ready to run the OSBL module of *naUTilus*. To do this, click on the 

button at the upper right hand portion of the screen.

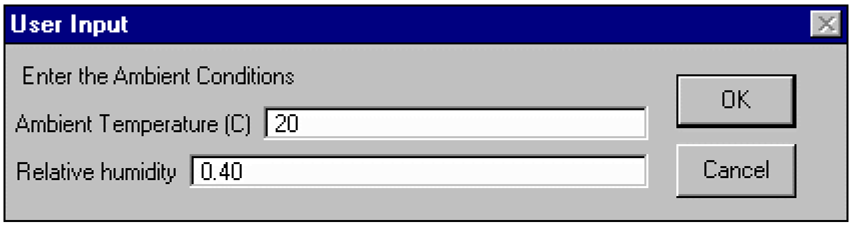
ArcView will prompt you to enter data on the number of pickholes per manhole cover and the area of the opening. Accept the default values and hit OK.



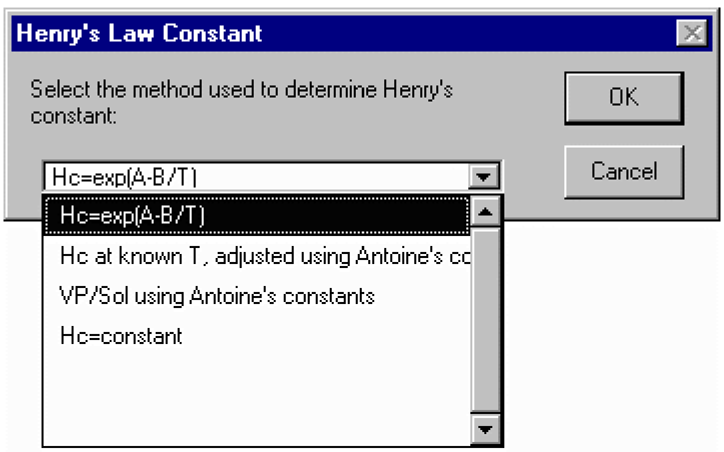
Next you will enter the ambient wind speed. Use a value of 1.2 meters per second.

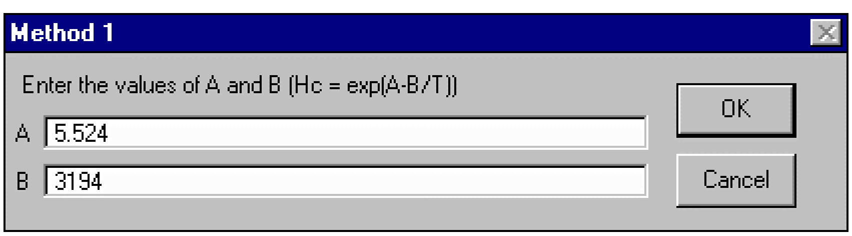


When prompted, enter a ambient temperature of 20 ^{o}C and a relative humidity of 40%.

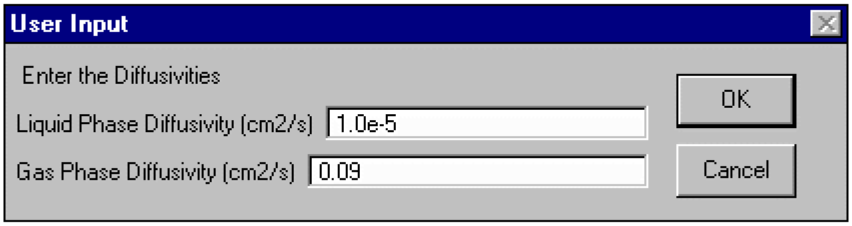


ArcView will then ask you to select the method of calculating the Henry's law constant. As for the ISBL unit, select method 1 (Hc = exp (A-B/T)). Enter the same values you input for the ISBL unit.

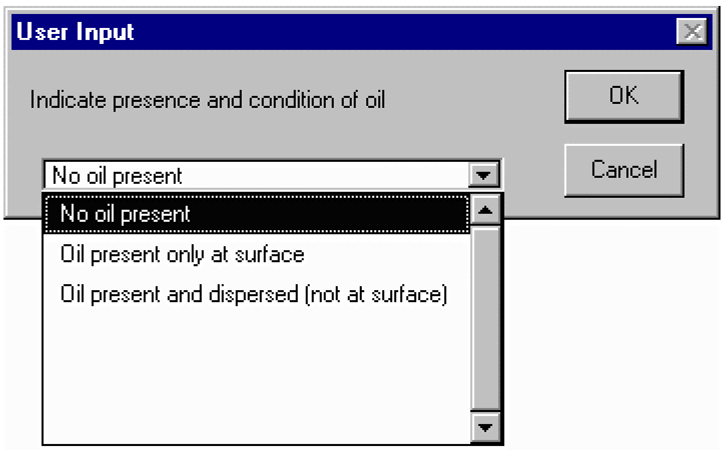


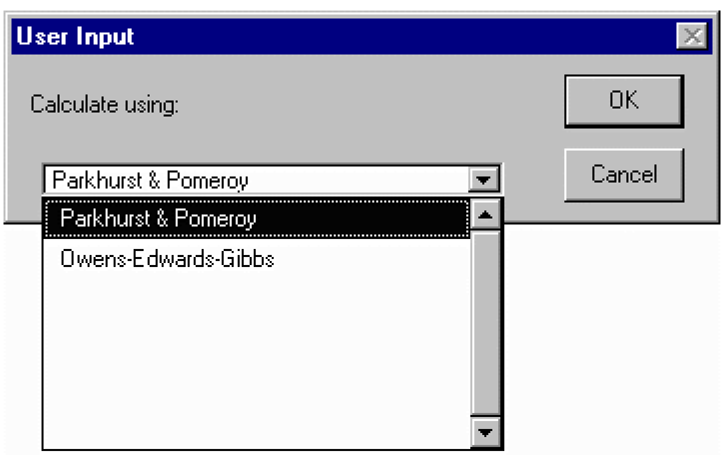


You will then enter the values for D_l and D_g. Again, use the same values you did for the ISBL unit.

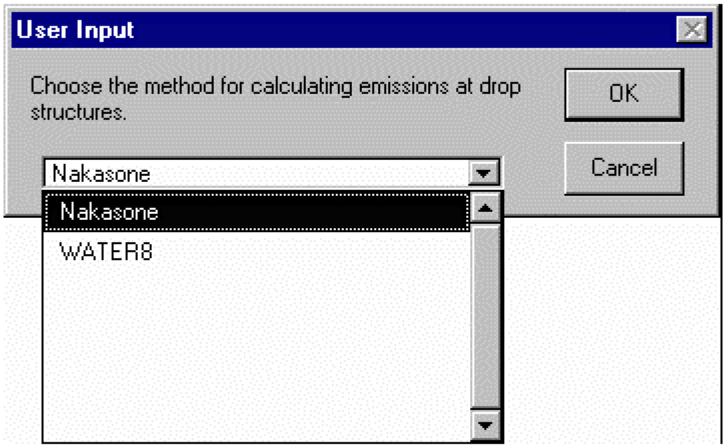


Again, you will use the same conditions for your OSBL as your ISBL when asked about the presence of oil in your system and the method of calculating liquid-phase mass transfer coefficients. Select No Oil Present and hit OK. Then select Parkhurst/Pomeroy and hit OK.

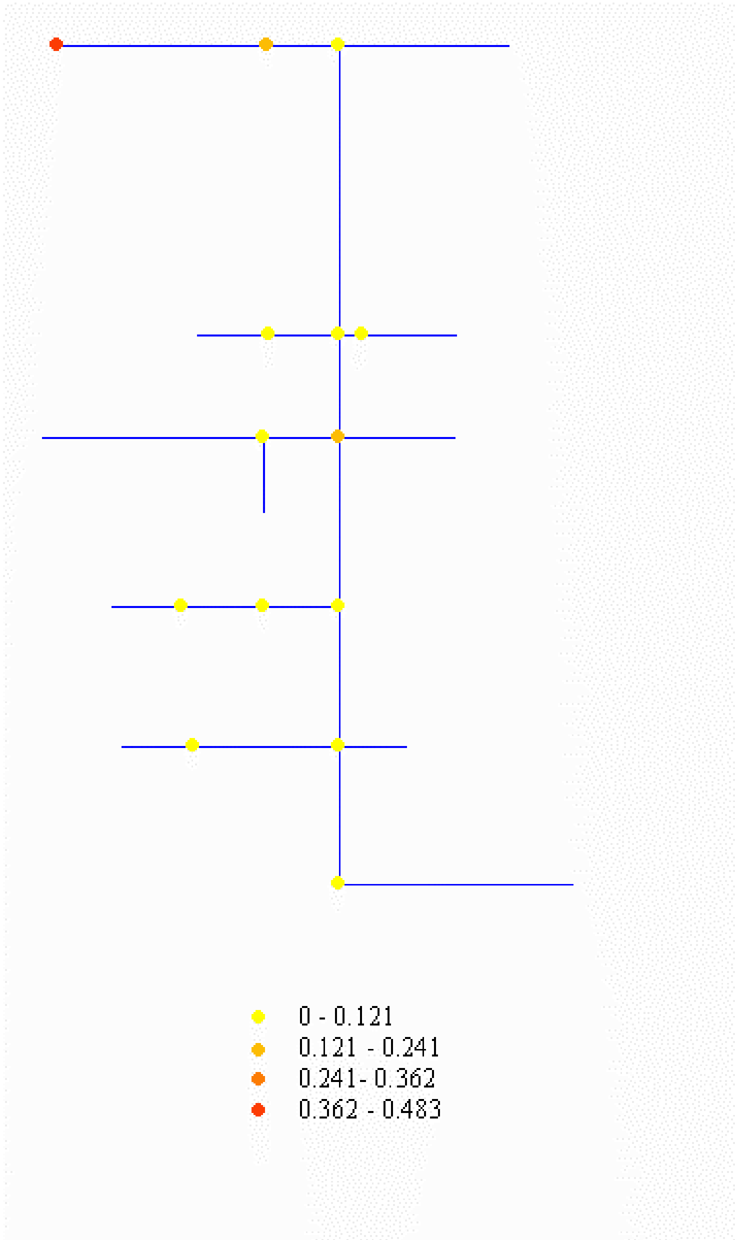




You will then select the method of calculating emissions at drop structures (Nakasone or Water8). Select Nakasone and hit OK.



ArcView will now run the OSBL module of *naUTilus*. Again, this is a delayed run, so about 5 seconds will pass before the program is run. As for the ISBL unit, a DOS window will appear when *naUTilus* is being run. After *naUTilus* has been executed, ArcView will display the emissions from the OSBL unit on the OSBL view window.



The OSBL module of *naUTilus* puts out three output files. Two are comma-delimited text files, one for nodes (ONDOUT.TXT) and one for branches (OBROUT.TXT). These files can be opened in ArcView to view the results. A third text file (OSBL.OUT) describes both nodes and branches and is best viewed in Notepad or word processing software such as Word.

ISBL.OUT:

```
---------------------------------------------
naUTilus

A Model for Predicting Chemical
Emissions from Industrial Sewers
                                                          version 1.0

developed by

David A. Olson, Sunil Varma, Richard L. Corsi

Program in Air Resources Engineering
The University of Texas at Austin
---------------------------------------------


 Network water flow rates for each branch (L/s)

 qibr(   1 )=     .000
 qibr(   2 )=    1.500
 qibr(   3 )=    1.500
 qibr(   4 )=    1.500
 qibr(   5 )=    1.000
 qibr(   6 )=    1.500
 qibr(   7 )=    1.000
 qibr(   8 )=    1.000
 qibr(   9 )=    1.500
 qibr(  10 )=     .000
 qibr(  11 )=    2.500
 qibr(  12 )=     .000
 qibr(  13 )=     .000
 qibr(  14 )=     .000
 qibr(  15 )=     .000
 qibr(  16 )=     .000

 Network water temperatures for each branch (C)

 tibr(   1)=   30.00
 tibr(   2)=   30.00
 tibr(   3)=   30.00
 tibr(   4)=   30.00
 tibr(   5)=   30.00
 tibr(   6)=   30.00
 tibr(   7)=   30.00
 tibr(   8)=   30.00
 tibr(   9)=   30.00
 tibr(  10)=   30.00
 tibr(  11)=   30.00
```

tibr(12)= 30.00
tibr(13)= 30.00
tibr(14)= 30.00
tibr(15)= 30.00
tibr(16)= 30.00

Network air flows for each branch (L/s)

qair(1)= .563
qair(2)= 3.539
qair(3)= 2.976
qair(4)= 2.976
qair(5)= 1.850
qair(6)= 2.413
qair(7)= 1.850
qair(8)= 2.413
qair(9)= 4.102
qair(10)= .563
qair(11)= 4.504
qair(12)= .563
qair(13)= .000
qair(14)= .563
qair(15)= .563
qair(16)= .563

Liquid and gas concentrations for each branch (mg/L)

1 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

2 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

3 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

4 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

5 Cli .9341 Clo .9115 eta 2.422%
Cai .0356 Cao .0478

6 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

7 Cli .9718 Clo .9341 eta 3.882%
Cai .0152 Cao .0356

8 Cli .9115 Clo .8883 eta 2.542%
Cai .0367 Cao .0463

9 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

10 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

11 Cli .3553 Clo .3501 eta 1.471%
Cai .0248 Cao .0277

12 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

13 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

14 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

15 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

16 Cli .0000 Clo .0000 eta .000%
Cai .0000 Cao .0000

Network totals

Above sewer emissions= .000 mg/s

Above sewer emissions (%)= .00
Overall stripping efficiency(%)= 64.99
Total mass entering the system = 2.50 mg/s
Emission rate= 1.62 mg/s

ISBLOUT.TXT:

Above_sew_emm,Tot_mass_in,Emission_rate,Flow_rate,Liq_conc,Temp,Oil_frac
.00000, 2.500, 1.62475, 2.500, .35010, 30.00, .00000

OSBL.OUT:

----

naUTilus

A Model for Predicting Chemical
Emissions from Industrial Sewers
version 1.0

developed by

David A. Olson, Sunil Varma, Richard L. Corsi

Program in Air Resources Engineering
The University of Texas at Austin

Network water flows (L/s)

```
Qbr(  1) =   2.500
Qbr(  2) =   2.500
Qbr(  3) =    .000
Qbr(  4) =    .000
Qbr(  5) =   2.500
Qbr(  6) =    .000
Qbr(  7) =    .000
Qbr(  8) =    .000
Qbr(  9) =   1.500
Qbr( 10) =   2.500
Qbr( 11) =   1.500
Qbr( 12) =    .000
Qbr( 13) =    .000
Qbr( 14) =    .000
Qbr( 15) =    .000
Qbr( 16) =   4.000
Qbr( 17) =    .000
Qbr( 18) =    .000
Qbr( 19) =   4.000
Qbr( 20) =    .000
```

Qbr(21) =   .000
Qbr(22) =   4.000
Qbr(23) =   4.000
Qbr(24) =   2.500

Network temperatures (C)

Tbr(1) = 30.000
Tbr(2) = 30.000
Tbr(3) =   .000
Tbr(4) =   .000
Tbr(5) = 30.000
Tbr(6) = 20.000
Tbr(7) = 20.000
Tbr(8) =   .000
Tbr(9) = 35.000
Tbr(10) = 30.000
Tbr(11) = 35.000
Tbr(12) =   .000
Tbr(13) =   .000
Tbr(14) =   .000
Tbr(15) = 20.000
Tbr(16) = 31.875
Tbr(17) = 20.000
Tbr(18) =   .000
Tbr(19) = 31.875
Tbr(20) = 20.000
Tbr(21) =   .000
Tbr(22) = 31.875
Tbr(23) = 31.875
Tbr(24) = 30.000

Network air flows (L/s)

Qgas(1)=   11.556
Qgas(2)=   11.556
Qgas(3)=   11.556
Qgas(4)=   11.556
Qgas(5)=   11.556
Qgas(6)=   .000
Qgas(7)=   .000
Qgas(8)=   11.592

Qgas(9)= 11.556
Qgas(10)= 11.592
Qgas(11)= 11.592
Qgas(12)= 11.592
Qgas(13)= 11.592
Qgas(14)= 11.592

~~~~~~
Emissions calculations : Equilibrium assumption
~~~~~~
Network concentrations (mg/L)

Cbr(1) = .1570
Cbr(2) = .0704
Cbr(3) = .0000
Cbr(4) = .0000
Cbr(5) = .0315
Cbr(6) = .0000
Cbr(7) = .0000
Cbr(8) = .0000
Cbr(9) = .2500
Cbr(10) = .0141
Cbr(11) = .2500
Cbr(12) = .0000
Cbr(13) = .0000
Cbr(14) = .0000
Cbr(15) = .0000
Cbr(16) = .0564
Cbr(17) = .0000
Cbr(18) = .0000
Cbr(19) = .0310
Cbr(20) = .0000
Cbr(21) = .0000
Cbr(22) = .0171
Cbr(23) = .0094
Cbr(24) = .3501

Mass emissions from node (mg/s) 1= .483
Mass emissions from node (mg/s) 2= .216
Mass emissions from node (mg/s) 3= .097
Mass emissions from node (mg/s) 4= .000
Mass emissions from node (mg/s) 5= .044

```
   Mass emissions from node (mg/s)  6=     .000
   Mass emissions from node (mg/s)  7=     .000
   Mass emissions from node (mg/s)  8=     .185
   Mass emissions from node (mg/s)  9=     .000
   Mass emissions from node (mg/s) 10=     .000
   Mass emissions from node (mg/s) 11=     .102
   Mass emissions from node (mg/s) 12=     .000
   Mass emissions from node (mg/s) 13=     .056
   Mass emissions from node (mg/s) 14=     .031

 ---------------------------------------------

 Network totals

 Total mass input to system (mg/s)=     1.250

   Total network emissions (mg/s)=     1.213

   Stripping Efficiency (%) =  97.00

 ---------------------------------------------
```

```
ONDOUT.TXT:
Node,Emissions
  1,  .4828650070
  2,  .2164741301
  3,  .0970479292
  4,  .0000000000
  5,  .0435077418
  6,  .0000000000
  7,  .0000000000
  8,  .1846528148
  9,  .0000000000
 10,  .0000000000
 11,  .1015622077
 12,  .0000000000
 13,  .0558609521
 14,  .0307244795
```

OBROUT.TXT:

```
Branch,Liq_flow,Temp,Gas_flow,Liq_conc
  1,   2.500,   30.00, 11.5559,  .15695
  2,   2.500,   30.00, 11.5559,  .07036
  3,    .000,     .00, 11.5559,  .00000
  4,    .000,     .00, 11.5559,  .00000
  5,   2.500,   30.00, 11.5559,  .03155
  6,    .000,   20.00,   .0000,  .00000
  7,    .000,   20.00,   .0000,  .00000
  8,    .000,     .00, 11.5919,  .00000
  9,   1.500,   35.00, 11.5559,  .25000
 10,   2.500,   30.00, 11.5919,  .01414
 11,   1.500,   35.00, 11.5919,  .25000
 12,    .000,     .00, 11.5919,  .00000
 13,    .000,     .00, 11.5919,  .00000
 14,    .000,     .00, 11.5919,  .00000
 15,    .000,   20.00,   .0000,  .00000
 16,   4.000,   31.88,   .0000,  .05643
 17,    .000,   20.00,   .0000,  .00000
 18,    .000,     .00,   .0000,  .00000
 19,   4.000,   31.88,   .0000,  .03104
 20,    .000,   20.00,   .0000,  .00000
 21,    .000,     .00,   .0000,  .00000
 22,   4.000,   31.88,   .0000,  .01707
 23,   4.000,   31.88,   .0000,  .00939
 24,   2.500,   30.00,   .0000,  .35010
```

Appendix F

ISBL SCHEMATICS

Random placement of sealed drains

The following images show examples of sealed drain placement used for random placement of sealed drains in the ISBL system. Each of the images represents two scenarios. The first has the sealed drain placement shown in the image. In the second scenario associated with the image, the sealed drain status of each drain is reversed (i.e. drains shown as sealed are unsealed and drains shown as unsealed are sealed). A summary of the scenarios, include those not shown here, is listed in Table F-1.

Scenarios 1 and 2:

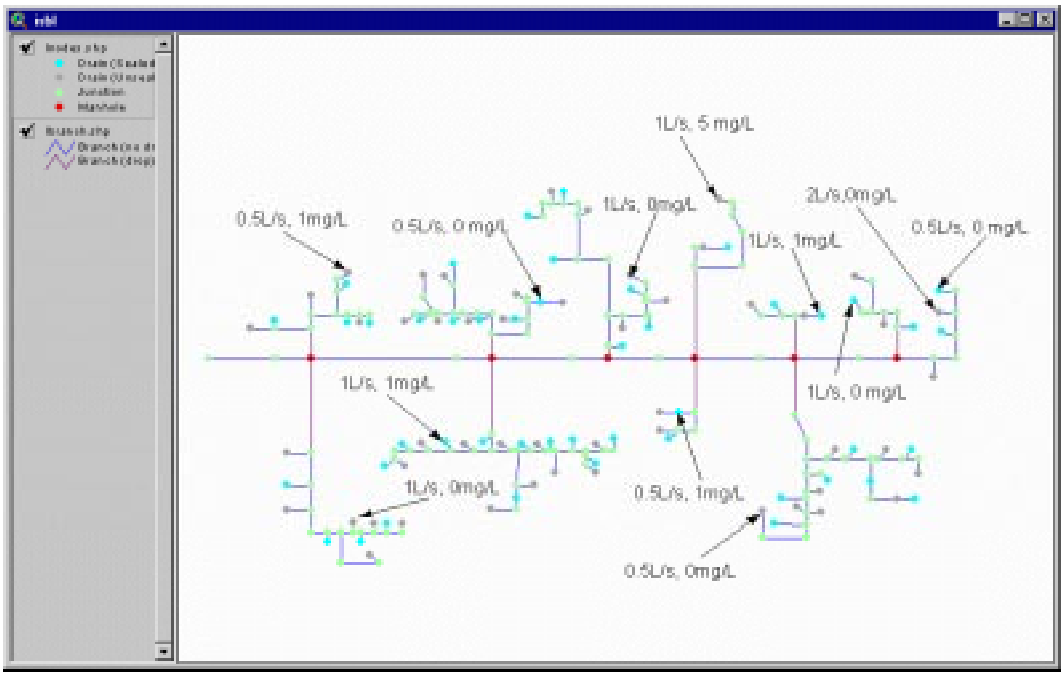


Figure F-1: ISBL unit with 46 sealed drains (random).

Scenarios 3 and 4:

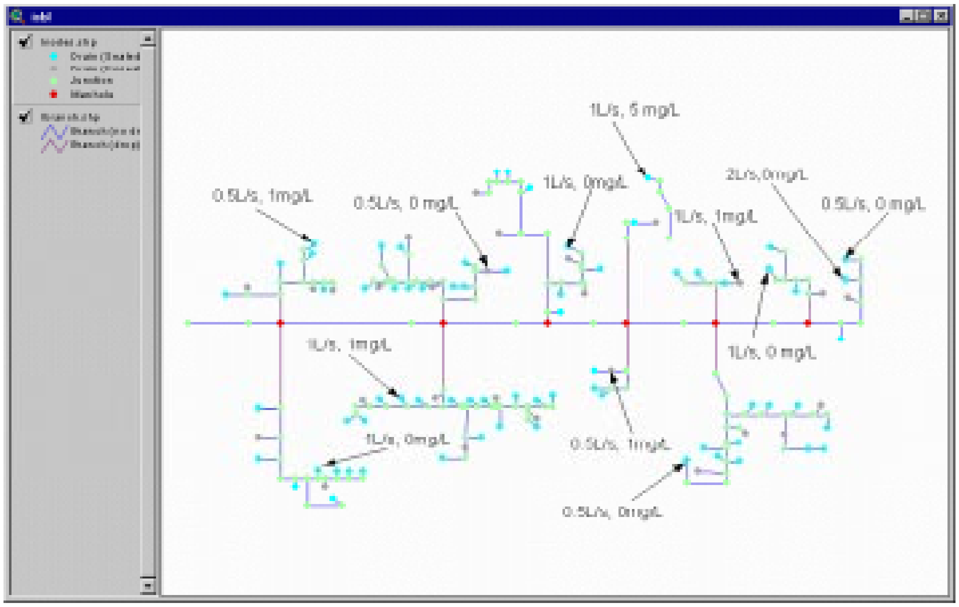


Figure F-2: ISBL unit with 72 sealed drains (random).

Scenarios 5 and 6:

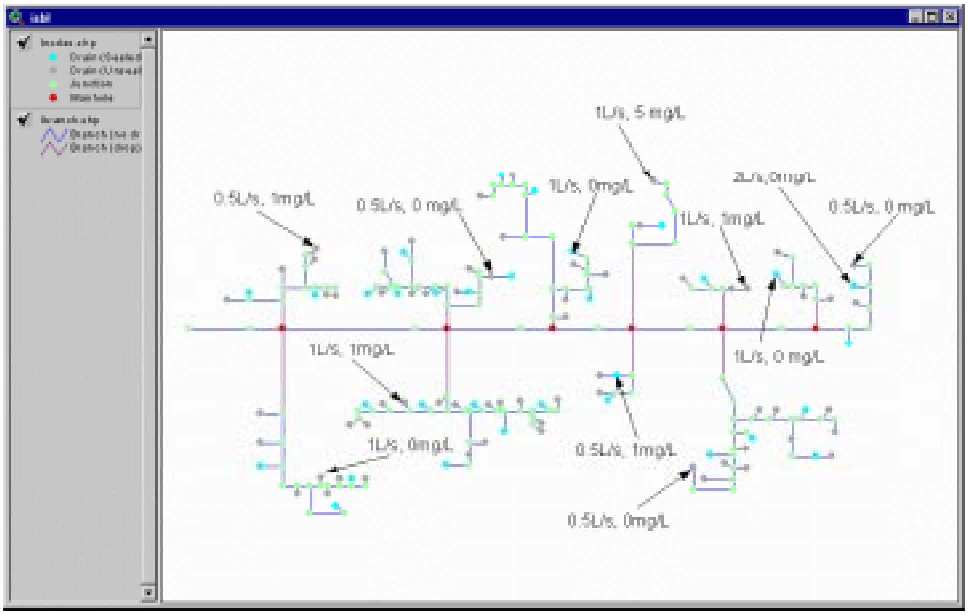


Figure F-3: ISBL unit with 32 sealed drains (random).

Table F-1: Summary of sealed drain scenarios for random drain placement.

Scenario #	Sealed drains	Unsealed drains	Sealed Drains w/flow
1	46	51	6
2	51	46	6
3	70	27	9
4	27	70	3
5	32	65	4
6	65	32	8
7	85	12	3
8	12	85	9
9	18	79	4
10	92	5	11

Grouped placement of sealed drains

The following images show an example of the sealed drain placement used for grouped placement of sealed drains in the ISBL system. Each of the images represents two scenarios. The first has the sealed drain placement shown in the image. In the second scenario associated with the image, the sealed drain status of each drain is reversed (i.e. drains shown as sealed are unsealed and drains shown as unsealed are sealed). A summary of the scenarios, including those not shown here, is listed in Table F-2.

Scenarios 1 and 2:

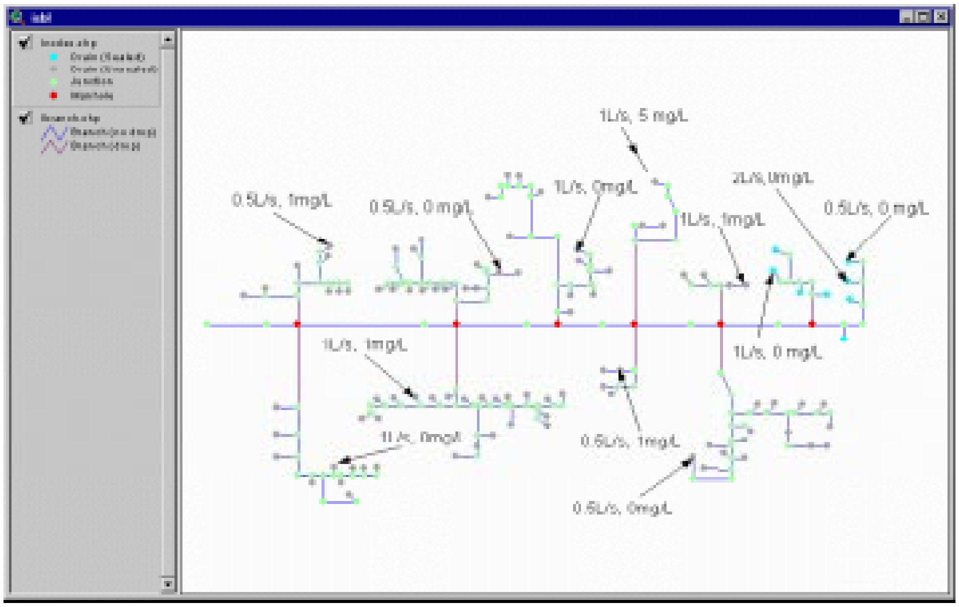


Figure F-7: ISBL with 8 sealed drains (grouped).

Scenarios 3 and 4:

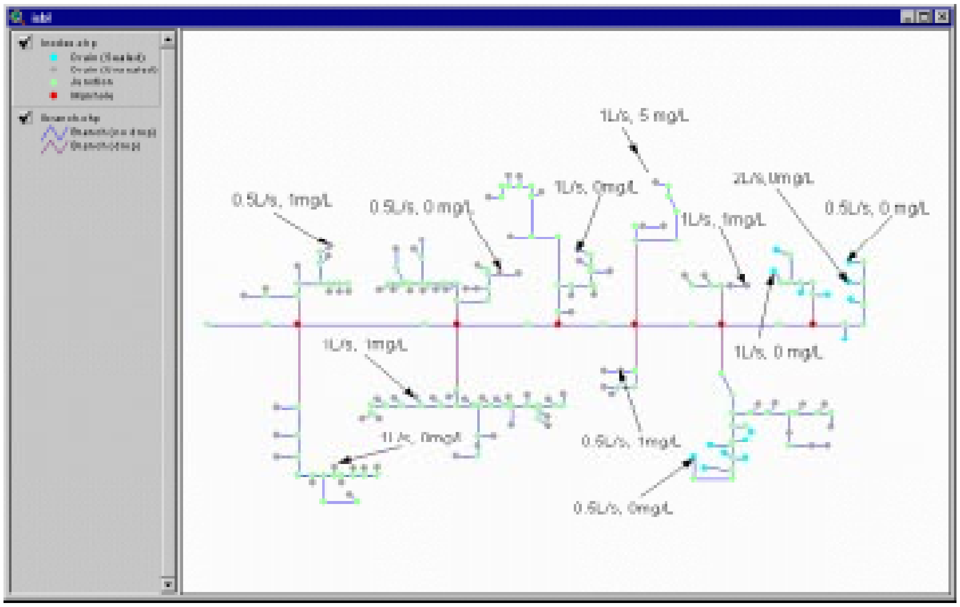


Figure F-8: ISBL with 15 sealed drains (grouped).

Scenarios 5 and 6:

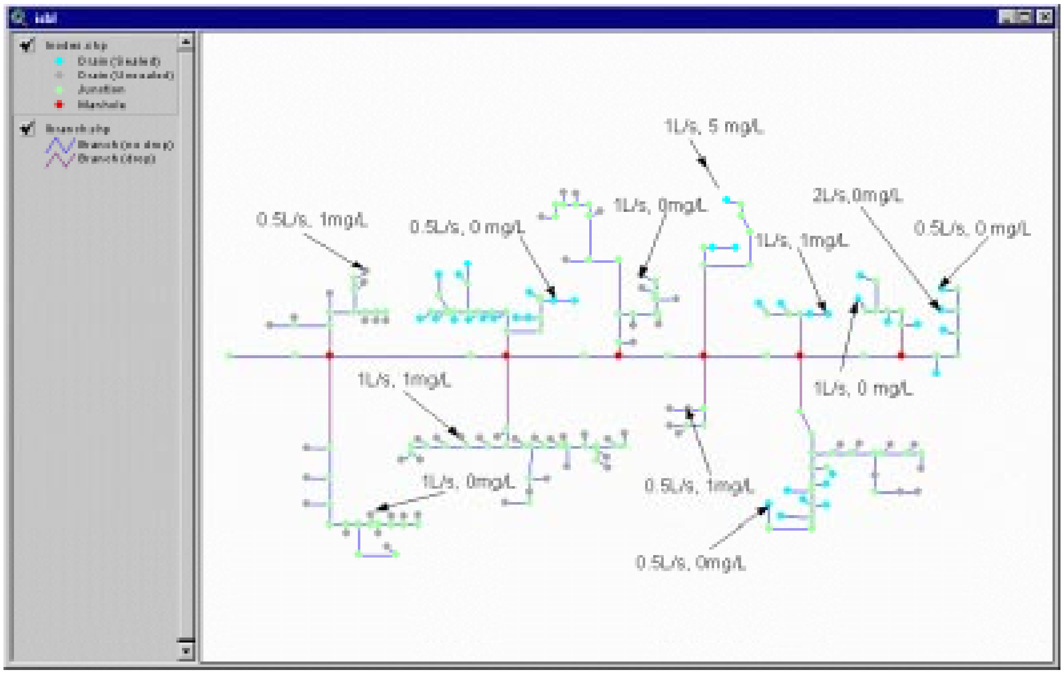


Figure F-9: ISBL with 36 sealed drains (grouped).

Table F-2: Summary of sealed drain scenarios for grouped drain placement.

Scenario #	Sealed drains	Unsealed drains
1	8	89
2	89	8
3	15	72
4	72	15
5	22	77
6	77	22
7	36	61
8	61	36
9	56	41
10	41	56
11	78	19
12	19	78

Hypothetical Inflows to OSBL

Four hypothetical ISBL units were assigned to the OSBL unit, along with 9 other flow sources. The four hypothetical ISBL units were based on the ISBL unit used throughout this research, with a variety of sealed drain conditions. Identical values were used for the inflow, ambient conditions, and chemical properties. The sealed drain conditions were as follows:

- Case 1: all drains unsealed.
- Case 2: all drains sealed
- Case 3: mixed (mostly unsealed)
- Case 4: mixed (mostly sealed)

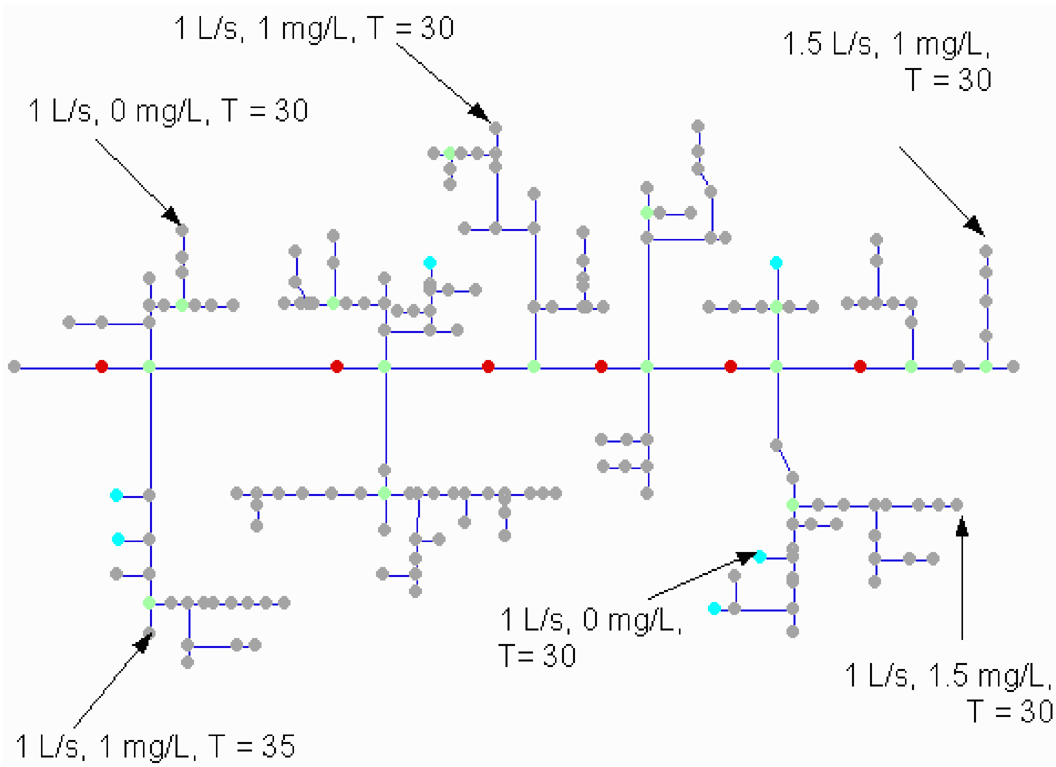


Figure F-10: ISBL unit, hypothetical flow Case 3 (mostly unsealed).

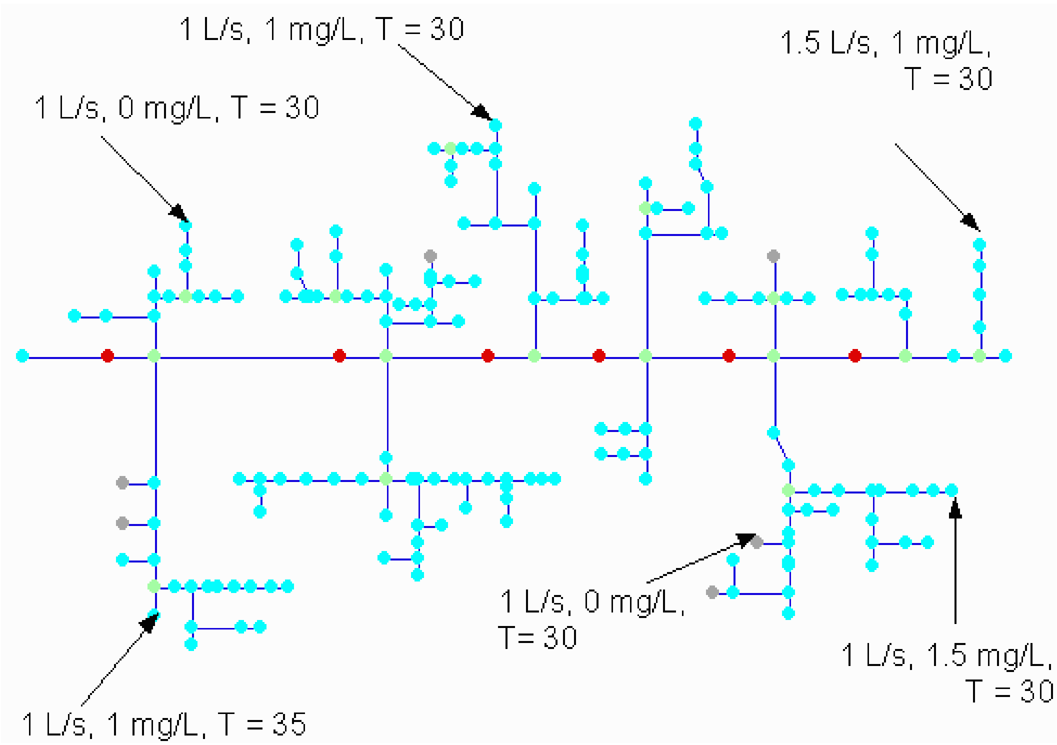


Figure F-11: ISBL unit, hypothetical flow Case 4 (mostly sealed).

References

American Petroleum Institute. 1996. Estimation of Fugitive Emissions from Petroleum Refinery Process Drains, Phase I Report.

Ashworth, R.A., G.B. Howe, M.E. Mullins, and T.N. Rogers. 1988. "Air-Water Partitioning Coefficients of Organics in Dilute Aqueous Solutions," *Journal of Hazardous Materials*, 18:25-36.

Benaman, J. 1996. *Modeling of Dissolved Oxygen in the Houston Ship Channel Using WASP5 and Geographic Information Systems*. M.S. Thesis, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas.

Berry, J.K. 1993 "Cartographic modeling: The Analytical Capabilities of GIS," in *Environmental Modeling with GIS*. Goodchild, M.F, B.O. Parks, and L.T Steyaert, *ed.*, Oxford University Press, New York, pp. 58-74.

Cano, Manuel. 1997. *Personal Communication*. Shell Devlopment Company, Houston, Texas.

Cooney, C.M. 1997. "Are Regulators Relying On Inaccurate Industrial Air Emission Inventories?" *Environmental Science & Technology*, 31(9):406A-407A.

Corsi, R.L., D.P.Y. Chang, and E.D. Schroeder. 1992 "A Modeling Approach for VOC Emissions from Sewers," *Water Environment Research*, 64(5): 734-741.

Corsi, R.L. 1996. *VOC Emissions from Industrial Wastewater: Estimation Methods*. A short course at The University of Texas at Austin, Austin, Texas.

Dean, J. A., *ed*. 1992. *Lange's Handbook of Chemistry*. 14th edition, New York: McGraw-Hill, Inc.

Douglas, W.J. 1995. *Environmental GIS: Applications to Industrial Facilities*. Boca Raton: Lewis Publishers.

Enviromega LTD. 1996. *SEAM Users' Manua*l Version 2.

Environmental Solutions Research Institute (ESRI). 1998. *ESRI Home Page*, World Wide Web Homepage, Address: http://www.esri.com.

Federal Register/ Vol. 59, No. 78 / Friday April 22, 1994, Rules and Regulations, HON Rule Appendix C to part 63.

Fedra, K. 1993. "GIS and Environmental Modeling," in *Environmental Modeling with GIS*. Goodchild, M.F, B.O. Parks, and L.T Steyaert, *ed.*, Oxford University Press, New York, pp. 35-50.

Hellweger, F. 1997. *Balance - Map Based Surface Water Quality Model*, World Wide Web Homepage, Address: http://www.ce.utexas.edu/prof/maidment/gishydro/ferdi/research/balance/balance.html.

Howle, R.H, and C.J. Zukor. 1994. *Low Emissions Sewer Systems for Industry*. Report to Water Environment Research Foundation.

Koziel, J.A. 1998. *VOC Emissions from Municipal Sewers: Hot Spots*. Ph.D. Dissertation, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas. In preparation.

Maitin, I.J. and K.Z. Klaber. 1993. "Geographic Information Systems as a Tool for Integrated Air Dispersion Modeling," Published in Proceedings: *GIS/LIS 1993 Annual Conference & Exposition,* ASCM, *et al.* Minneapolis, Minnesota.

Mackay, D. and T.K. Yeun. 1983. "Mass Transfer Coefficient Correlations for Volatilization of Organic Solutes from Water," *Environmental Science & Technology*. 17, 211.

Moeller, R.A. 1998. *Personal Communication*. Montgomery Watson, Baton Rouge, Louisiana.

Nakasone, H. 1986. "Study of Aeration at Weirs and Cascades," *Journal of Environmental Engineering*, ASCE 113:1, 64.

Nobel, C. 1998. *An Industrial Ecology-Based Model for Water Reuse: A Geographic Information Systems (GIS) Approach*. M.S. Thesis, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas. In preparation.

Novak, J.H. and R.L. Dennis 1993. "Regional Air Quality and Acid Deposition Modeling and the Role fo Visualization," in *Environmental Modeling with GIS*. Goodchild, M.F, B.O. Parks, and L.T Steyaert, *ed.*, Oxford University Press, New York, pp. 142-146.

Nyerges, T.L. 1993. "Understanding the Scope of GIS: Its Relationship to Environmental Modeling," in *Environmental Modeling with GIS*. Goodchild, M.F, B.O. Parks, and L.T Steyaert, *ed.*, Oxford University Press, New York, pp. 75-93.

Olson, D.A. 1996. *Gas Exchange Rates Between Industrial Process Drains and the Ambient Atmosphere*. M.S.Thesis, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas.

Olson, D.A., R.L. Corsi, and S. Rajagopalan. 1996. "Buoyancy-Induced Ventilation between Industrial Sewers and the Ambient Atmosphere," Published in proceedings: *89th Annual Meeting of the Air & Waste management Association*, A&WMA, Nashville, Tennessee.

Olson, D.A., S. Rajagopalan, and R.L. Corsi. 1997(a). "Ventilation of Industrial Process Drains: Mechanisms and Effects on VOC Emissions," *Journal of Environmental Engineering*, ASCE 123(9):939-947.

Olson, D.A., S. Varma, and R.L. Corsi. 1997(b). *naUTilus: A Model for Predicting Chemical HAP Emissions from Industrial Sewers*; I. User Documentation, II. Technical Documentation, report to the United States Environmental Protection Agency.

Parkhurst, J.D. and R.D. Pomeroy. 1972. "Oxygen Absorption in Streams," *Journal of Sanitary Engineering Division*, ASCE 98, 101.

Stubbe, J.K. 1997. *VOC Emissions from Industrial Process Drains: Mechanistic Model and Parameter Estimation*. M.S. Thesis, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas.

Tim, U.S. and R. Jolly. 1994. "Evaluating Agricultural Nonpoint-Source Pollution Using Integrated Geographic Information Systems and Hydrologic/Water Quality Model," *Journal of Environmental Quality*, 23: 25-35.

U.S. Environmental Protection Agency (USEPA). 1990. *Industrial Wastewater Volatile Organic Compound Emissions -- Background Information for BACT/LAER Determinations*, Office of Air Quality Planning and Standards.

U.S. Environmental Protection Agency (USEPA). 1992. *Hazardous Air Pollutant Emissions from Process Units in the Synthetic Organic Chemical Manufacturing Industry -- Background Information for Proposed*

Standards, vol. 1C, Model Emissions Sources. Office of Air and Radiation.

U.S. Environmental Protection Agency (USEPA). 1994a. *Air Emissions Models for Water and Wastewater*. Office of Air Quality Planning and Standards.

U.S. Environmental Protection Agency (USEPA). 1994b. *User's Guide for Wastewater Treatment Compound Property Processor and Air Emissions Estimator (WATER8)*. Office of Air Quality Planning and Standards.

U.S. Environmental Protection Agency (USEPA). 1995. *Compilation of Air Pollutant Emission Factors, AP-42*, Fifth Eddition, Volume 1, Statonary Point and Area Sources.

U.S. Environmental Protection Agency (USEPA). 1998. *Emissions Estimation Software*. World Wide Web Homepage, Address: http://www.epa.gov/ttnchie1/software.html.

Varma, S. 1995. *VOC Emissions from Industrial Sewers: Source Characterization and Passive Control*. M.S. Thesis, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas.

Whitmore, A., R.L. Corsi. 1994. "Measurement of Gas-Liquid Mass Transfer Coefficients for Volatile Organic Compounds in Sewers," *Environmental Progress*, 13: 114-123.

Ye, Z. 1996. *Map-based Surface and Subsurface Flow Simulation Models: An Object-Oriented and GIS Approach*. Ph.D. Dissertation, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas.

Vita

Cindy How was born in Winnipeg, Manitoba, Canada, on March 28, 1974, the daughter of Jeannette Foong Ching How and George That Sen How. After moving to Calgary, Ontario in 1978, she and her family relocated to Mililani, Hawaii in 1980. After graduating from Mililani High School in 1992, she attended Tulane University in New Orleans, Louisiana as a Dean's Honor Scholar. While attending Tulane, she was a summer intern at Oak Ridge National Laboratories, working in the Environmental Science Division. She also studied at the Univeristy of Birmingham, in Birmingham, England, through a Tulane Junior Year Abroad program. She completed her senior year at Tulane University, graduating in May 1996 with a Bachelor of Science degree in Environmental Engineering, *summa cum laude*. After graduating, she worked as a summer intern, and later as a consultant, in the Civil and Environmental Engineering department at Tulane University, conducting an environmental justice study in geographic information systems. She entered graduate school at The University of Texas at Austin in August 1996.

Permanent address: 94-160 Keaolani Street
Mililani, Hawaii 96789

This thesis was typed by the author.